

Sinclair COMPUTER

mensile per gli utenti dei computer Sinclair

#16 Lire 3000
settembre 1985

Il data-base



Systems





L. 12.000

presentano

**Sinclair
COMPUTER**

DAN·PICVS e MIRANDVLA



**BANCA
DATI**

VERSIONI PER

COMMODORE 64

&

SPECTRUM

Systems

Il tuo
DATA BASE
personale

Suppl. a Sinclair Computer n. 16 - 1985 - Distr. MePe



- 04 - Sinclairmente vostro/la posta
- 08 - software: La Torre di Hanoi
- 10 - il cruciverba del n.15
- 11 - software: Poker
- 12 - software: Window
- 13 - software: Monoscopio
- 15 - spazio QL/Utility
- 16 - spazio QL/Posta
- 18 - spazio QL/Superbasic
- 20 - Il DATABASE
- 23 - Trasferimenti al microdrive
- 26 - recensioni: i libri
- 29 - I canali dello Spectrum
- 30 - Un po' di PASCAL
- 37 - Assembly con lo Spectrum
- 39 - I movimenti sullo schermo
- 40 - Hard-copy doppia grandezza
- 42 - recensioni: i videogames
- 45 - Del monitor
- 46 - Un tasto per volta
- 48 - La mappa dello schermo
- 51/64 - I LISTATI
- 65 - Piccoli annunci

SINCLAIR
COMPUTER

REDAZIONE
Mauro Soldavini, Fabio Bero, Marco De Martino

SEGRETARIA DI REDAZIONE
Maura Ceccarelli, Piera Perin

COLLABORATORI
Paolo Benvenuti, Marco Bertani, Carlo Bolchini, Damiano Bolla, Giuliano Boscchi, Luigi Callegari, Gianluca Carri, Valerio Cipolletti, Paolo Dray, Fabrizio Ferrario, Monica Fumagalli, Guido Grassi, Giovanni Mellina, Marco Musasini, Roberto Previtera, Renzo Zorini

PROVA E IMPAGINAZIONE
Guglielmo

DIFFUSIONE E ABBONAMENTI
Marina Vantini

DIREZIONE, REDAZIONE
Viale Farnagosta 75 - 20142 Milano -
Tel. (02) 8467348/9/40

PUBBLICITÀ
Milano: Mirco Croce (coordinatore), Giuseppe Porzani, Michela Prandini, Giorgio Ruffoni, Claudio Tidone, Villa Claudio
Segreteria: Liliana Degiorgi
- Via Farnagosta 75, 20142 Milano - tel. (02) 8467348/9/40
Roma: Spazionzero di R. De Marinis via P. Foccarli 70, 00139 Roma tel. (06) 8109679

FOTOCOPOSIZIONE
Fotocomposizione LM (Brescia)

STAMPA
La Litografica S.r.l. (Busto Arsizio)

DISTRIBUZIONE
Messaggerie Periodici S.p.A.
via G. Carcano 32, Milano
Spedizione in abb. Post. GR. 8/70

SYSTEMS EDITORIALE s.r.l.
(Registro Nazionale Stampa
n. 01500 vol. 15 fg. 793)

Direttore responsabile: Agostina Ronchetti
Autorizz. Trib. di MI n. 255/12.11.1983

Una copia L. 3.000 (Arretrati L. 6.000)
Abbonamento annuo (11 numeri) L. 28.000 (estero il doppio). I versamenti e le richieste di arretrati vanno indirizzati a: Sinclair Computer, Via Farnagosta 75, 20142 Milano, mediante emissione di assegno bancario o versamento sul c/c postale n. 37952207 intestato a SYSTEMS editoriale. Per i cambi di indirizzo indicare, unitamente al nuovo, anche l'indirizzo precedente, allegando L. 500 in francobollo.

Sinclair ZX81, ZX Spectrum, ZX Microdrive, QL sono marchi registrati della Sinclair Research Ltd.



sinclair *amente vostro*

Lo Spectrum plus

Per tutti coloro che scrivono manifestando dubbi: lo Spectrum nella nuova tastiera "plus" è totalmente compatibile con qualsiasi interfaccia, espansione, joystick, altoparlante, drive, etc., nonché con tutto il software preesistente, che funzionasse su un normale 48K, in quanto è ancora un normale 48K, a cui è stato sostituito esclusivamente l'involo.

Come si può collegare lo Spectrum a un monitor con ingresso RGB? Per collegare un monitor a colori occorre un'interfaccia? Ci si può collegare alla presa VCR di un televisore? (vari lettori)

Per l'ennesima volta: lo Spectrum non è collegabile a un monitor con ingresso RGB; il collegamento è possibile soltanto agli apparecchi con ingresso composito. In parole povere, il segnale deve essere tutto su una linea, mentre l'RGB ha tre linee separate (la sigla sta infatti per Red, Green, Blue, i colori base). Non occorre interfaccia di sorta, ma è meglio prendere qualche precauzione (vedi l'articolo sul monitor in questo numero). Ancora: gli ingressi VCR (per videoregistratore) dei televisori bypassano il demodulatore, usando il tv come monitor, quindi vanno benissimo. I connettori SCART sono illustrati nel suddetto articolo.

Come si carica il primo programma di SC n. 9, e in genere quelli in assembler? Perché nei listati non compaiono ORG e END? Come si disattiva il tasto BREAK? (R. Ravani - Monza MI)

I programmi in linguaggio macchina si caricano con i ben noti "caricatori", o hex-loader, pubblicati da noi e da tutti i libri o riviste in un'infinità di versioni, se scritti in esadecimale; avendo a disposizione le istruzioni in assembly, si può trascrivere il programma con un compilatore del linguaggio (un assembler, appunto). Nei nostri listati non compaiono, nella maggioranza dei casi, ORG e END, perché queste sono direttive specifiche del compilatore (nemmeno unificate), e non fanno parte del set di istruzioni del microprocessore Z80; inoltre, in genere pubblichiamo listati prodotti disassemblando il codice macchina, e in questo caso non esiste l'ORG - che, peraltro, è semplicemente la prima locazione occupata dal programma stesso.



È possibile sviluppare un 42 colonne a caratteri non rimpiccioliti da qualche word-processor? Esiste

un'interfaccia per portare il suono del beep simile a quello del C64? (T. Monti - Cusano MI)

Lo Spectrum produce suoni con una singola "voce", il C64 possiede un integrato (esclusivo della Commodore) specifico per la generazione di suoni complessi: la differenza è incolmabile, non ci risulta che esistano espansioni hard o soft che diano risultati paragonabili a quelli del concorrente. Quanto alla prima domanda, poiché l'ingombro di un carattere in larghezza non supera i 6 pixel, e $6 \times 42 = 252$, si possono effettivamente avere 42 caratteri per riga senza rimpicciolimento (sia pur ottenendo le maiuscole completamente accostate). I word-processor in circolazione utilizzano tutti un sistema a 64 colonne.

RANDOMIZE USR 200 è uguale a RUN? Che cosa significa la scritta che compare dopo RAND USR 9? Quando bisogna dare POKE per linguaggio macchina in hex e quando in dec? Quali sono gli effetti di OUT sul BORDER? (Calvi - Genova)

1. Per niente: la locazione 200 si trova in mezzo alla tabella dei token, e chiamando IF il computer si inchioda.

2. Niente: il restart della routine di errore è USR 8.

3. In hex, mai: POKE riconosce solo argomenti decimali.

4. La cornice "risponde" alla porta logica 254, perciò eseguendo



OUT 254,n (con n entro il corretto range dei colori) si cambia il BORDER.

A volte, salvando o caricando con il microdrive, appare l'OK, ma il nastro continua a girare, o bisogna dare un BREAK; nemmeno questo funziona sempre, e devo togliere la corrente.

Si può collegare lo Spectrum a un Olivetti M10 attraverso l'interfaccia 1? (M. Anchieri - Borgomanero NO)

Alzi la mano chi, possessore di microdrive, non è mai incorso lamentato da questo lettore. Anche se la cosa è molto sgradevole, nella maggior parte dei casi togliendo la corrente si perde solo il programma in memoria, mentre la cartuccia si salva. Se però succede molto spesso, qualcosa non va; anzitutto, prova a pulire con delicatezza i contatti dei vari pettini di collegamento, evitando di toccarli con le dita: devono essere perfettamente sgrassati; se il malfunzionamento persiste, la periferica richiede un intervento specializzato.

L'Olivetti M10 può essere collegato allo Spectrum per lo scambio di dati, mediante la porta RS-232: usando il programma TELCOM (residente nell'M10) non si riesce però a superare la velocità di 600 baud,

probabilmente per carenze della RS-232 di questo computer.

Non riesco a decidere quale stampante acquistare... come si stampa a 80 colonne con lo Spectrum? Che cosa significa stampante grafica? è quasi un plotter? (R. Pescini - Piombino)

La stampa a 80 colonne, o qualsiasi altro numero, dipende dalla stampante, non dal computer: qualsiasi stampante che si rispetti ha dei codici di controllo che "formatano" (orrendo vocabolo, ma ormai ineliminabile) la pagina, stabilendo tra l'altro margine destro e sinistro e di conseguenza il numero di colonne da utilizzare. La presenza di opzioni grafiche, riferendosi ai tipi di stampante più diffusi, cioè quelli ad aghi, consente, per esempio, di effettuare copie di disegni dallo schermo, ma non ha niente a che vedere con i plotter, in cui la funzione principale è il disegno, e quella di scrittura è solo accessoria.

Avendo in esecuzione il seguente programma, si rientra al basic premendo contemporaneamente A,S,L: perché? (L. Lomi - Torino)

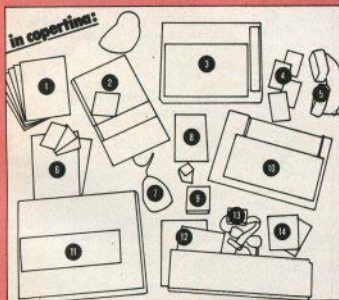
LD A,BFh
IN A, Feh
BIT 0,A
JR NZ,F8h

RET

Il ritorno al basic si ha anche con altre combinazioni di tasti (tra cui semplicemente ENTER): in carica l'accumulatore con il valore fornito dalla porta 254, BIT ne testa il bit 0, e si ha il RET se ne risulta il flag zero a 0. Quando si preme un tasto si producono due segnali, uno dei quali va appunto alla porta 254, e può quindi essere letto dal processore.

Ho sentito parlare di speciali filtri da inserire nello Spectrum tramite le prese EAR e MIC collegati con due registratori per ottenere la massima perfezione nella registrazione anche di quelli denominati "turbo". È vero? (S. Zardo - Ubaldo VA)

Mettiamola così: i duplicatori professionali, quelli che, legalmente o no, producono grandi masse di cassette da mettere sul mercato, usano impianti ad alta velocità, dotati di (costosissimi) sistemi di trattamento del segnale, paragonabili concettualmente ai riduttori di rumori tipo Dolby degli stereo; è possibile che qualcuno si sia dato la briga di assemblare una versione economica di tali circuiterie, per dotare i copiatori domestici di un ammenicolo in più. Se anche costui esiste, non ne conosciamo l'indirizzo.



1. collana di libri della McGraw-Hill dedicata allo Spectrum
2. unità integrata (con disk-drive da 3.5 pollici) Opus Discovery1 (Bit Shop Primavera)
3. stampante JP-80 (Telcom)
4. videogames Mastertronic
5. accoppiatore acustico Sendata
6. software gestionale McGraw-Hill per Spectrum
7. joystick "the Arcade" (Mastertronic)
8. Gyron, avventura grafica per Spectrum abbinata a un concorso; primo premio una Porsche 924 (Fi-rebird)
9. microdrive e cartridges
10. stampante Seikosha SP-800 (Rebit)
11. stampante Epson FX-80 (Epson/SEGI)
12. volumi sul 68000 e sul sist. operativo del QL
13. interfaccia seriale/parallelo per QL
14. software per QL



Come eseguire la traduzione per lo Spectrum del programma **Calendari**, destinato allo ZX81, apparso sul n. 2? (A. Bondoni - Firenze)

Il programma può essere trascritto così com'è, relativamente a tutta la parte di calcolo; si deve solo togliere o sostituire la routine di ingrandimento caratteri (la stringa alla linea 8020 fa parte di questa routine).

Esiste una routine che salvi su nastro solo una finestra dello schermo? (Davide Zenna)

Proprio in questo numero pubblichiamo un programma di un lettore per trasferire in RAM una finestra dello schermo: da qui il blocco di bytes può essere salvato; ovviamente dovrà essere riportato sul display file dalla stessa routine.

Vorrei una mappa per gli indirizzi in assembly e lo schema elettrico dell'hardware dello Spectrum. (G. Mandrà - Brescia)

Lo schema elettrico può essere fornito da chi svolge assistenza tecnica. La "mappa degli indirizzi in assembly" è un'espressione ambigua, ma supponiamo che tu riferisca alle routines della ROM: consigliamo il libro "Tutti i segreti dello Spectrum", edito da noi e opera del nostro collaboratore G. Mellina. Si trova nelle edicole o presso di noi, a L. 7000 + spese di spedizione.

Come funzionano le chiamate RST alla "pagina zero"? In particolare, come si usa RST 40dec? Quali sono i codici possibili? C'è un libro che illustra le routines utilizzabili della ROM? Come trasformare VAL in assembly? (B. Inceprona - Comiso)

L'istruzione RST (restart) funziona come un CALL, ovvero una USR da basic, ma occupa meno memoria. Per una trattazione esauriente consigliamo anche in questo caso il nostro libro sul sistema operativo (cfr. risposta precedente), che contiene uno specifico capitolo sul calcolatore in virgola mobile. I codici utilizzabili da RST 40, i cosiddetti literals, sono ben 82, e non possono venire affrontati in poche righe. VAL è eseguita dal literal 29.

Problema: disegnato un cerchio sullo schermo, "riempito" di colore, passando con il cursore con un diverso INK sul cerchio non si muove un puntino, ma un quadrato grande come un carattere, che non lascia una coda sottile di pixel, ma grossa quanto, appunto, un carattere. Si può risolvere il problema? (G. Santoro - Francavilla ME)

Sullo Spectrum non è possibile sovrapporre due INK diversi nel medesimo carattere, poiché l'attributo relativo a INK è uno solo per ciascuno quadratino di 8*8 pixel. Per evitare questi inconvenienti occorre combinare PAPER e INK in modo che non interferiscano, ma spesso è impossibile: anche nei giochi più sofisticati il problema rimane, anche se "nascosto" dalla velocità dei movimenti.

È possibile aggiungere a uno Spectrum 48k i nuovi comandi dello Spectrum plus? (E.C. Lucerna - Balangero TO)

Certamente... dato che non esistono: lo Spectrum plus ha dei TASTI in più, ma nessun comando nuovo.

Come si ridefinisce il set non grafico dei caratteri? Come si ottengono colori non presenti in tastiera? (D. Ballerini - Prato)

Il set di caratteri standard risiede in ROM, ed è puntato dalla variabile di sistema CHARS (23606/7). Volendo usare caratteri diversi, si disegnano allo stesso modo degli UDG e si memorizzano in RAM, abbassando RAMTOP, come per bytes di linguaggio macchina; quindi si modifica CHARS con due POKE, scrivendovi l'indirizzo da cui inizia il nostro alfabeto, meno 256: questo perché i primi 32 sono caratteri non stampabili. Tenete presente che dovete trascrivere l'intero set, da CHR\$ 32 a CHR\$ 127.

Non si possono avere colori non presenti in tastiera, ma si può ricorrere a qualche truccetto per avere effetti cromatici più completi; per esempio ponendo su un fondo (PAPER) blu una quantità variabile di pixel a INK giallo, disposti a scacchiera, si avranno toni diversi di verde.

Vorrei acquistare una tavoletta grafica. Esistono plotter economici? (A. Rabino - Torino)

Esistono tavolette grafiche dotate di software specifico per lo Spectrum: il mercato inglese offre sicuramente una maggiore varietà di prodotti. Il plotter più economico (in realtà, un ibrido tra plotter e stampante) è il Tricom CPP 114, presentato sul n. 13. Per un vero plotter, che disegni almeno sul formato A4, il concetto di economico è relativo... anche se stanno arrivando sul mercato prodotti con ottime prestazioni, che solo due anni fa erano riservate a macchine professionali. Non abbiamo marche privilegiate di trasformatori, l'importante è che diano corrente e tensioni sufficienti e costanti.

In previsione del collegamento di periferiche, è sufficiente un alimentatore da 2 ampere? (E. Vitaliano - Roma)

Sicuramente sì. Anche perché, nel caso si volessero collegare periferiche con un assorbimento maggiore (peraltro improbabile), consiglieremo per prudenza di dotarle di alimentazione separata.

Un amico commodorian possiede un simulatore di Spectrum per il C64: esiste qualcosa che fa l'inverso? (A. Bianchi - Sanremo)

Esiste, sarebbe prodotto dalle stesse persone (tedesche) che hanno fatto quello per il C64; al momento in cui scriviamo, non ne siamo ancora venuti in possesso; speriamo di potervi dire di più molto presto.

In un programma per lo ZX81 ho usato queste istruzioni:

```
INPUT Y$
LET Y = INT (VAL Y$)
Forndo in input una potenza, come X**2 + 3, il computer va in errore: perché? (L. Bonaventura - Corsico MI)
```

L'unica spiegazione che riusciamo a trovare è che per scrivere il segno di elevamento a potenza (un doppio asterisco per lo ZX81), tu abbia battuto proprio un doppio asterisco, anziché il simbolo ottenibile con SHIFT + H.



NELLE TUE MANI

tutta la potenza di una grande stampante

P-40 ideale per home e personal computer

Questa è Epson P-40, la stampante termica ultracompatta, quasi tascabile, la compagna ideale per il tuo personal computer a casa, a scuola e anche nel lavoro.

Piccola, robusta, progettata per lavorare a lungo e realizzata con la proverbiale qualità Epson, la P-40 funziona con batterie ricaricabili e stampa grafici e testi su 20, 40 o 80 colonne (modo compresso) a 45 caratteri al secondo.

Regala Epson P-40 al tuo personal. Con la piccola Epson il tuo personal diventa grande!

P-80

P-80 e P-80X la qualità di stampa professionale

Con la nuova P-80 e il tuo personal computer hai la stessa qualità delle stampanti a matrice da tavolo a 80 colonne per produrre prospetti proposte d'acquisto, tabelle o listini di elevata qualità su carta termica o su carta comune. Se poi desideri una qualità di stampa virtualmente indistinguibile da quella delle macchine da scrivere, scegli P-80X, con i suoi 24 "aghi" capaci di produrre caratteri pieni e netti, autorevoli, per la tua corrispondenza più importante. Quando vuoi, dove vuoi.

P-80 e P-80X stampano su 40, 80 e 136 colonne su fogli singoli a 45 caratteri al secondo.



P-40

EPSON l'informatica portatile, anche nelle periferiche



Sulla torre di Hanoi abbiamo già presentato un programma per Spectrum, sul n. 10, in cui la soluzione doveva essere trovata dal giocatore. Come avete potuto constatare, più è alto il numero di dischi scelto, più la soluzione è difficile da trovare, maggiore, in crescita geometrica, il numero di mosse e, di conseguenza, più alto il tempo impiegato, che pure, supponendo di spostare ogni disco in un tempo costante, aumenta esponenzialmente. Qui forniamo due programmi, uno in basic, uno in linguaggio macchina, che suggeriscono la soluzione in un batter d'occhio, e per un numero di dischi fino a 19 (!!).

Ma non si tratta solo di fornire la soluzione a un gioco (sarebbe un modo un po' squallido di riempire le pagine): la torre di Hanoi costituisce un esempio classico, ormai proverbiale, di programmazione avanzata, su cui vale veramente la pena di ponderare un po', se si vuole alzare la qualità dei propri programmi.

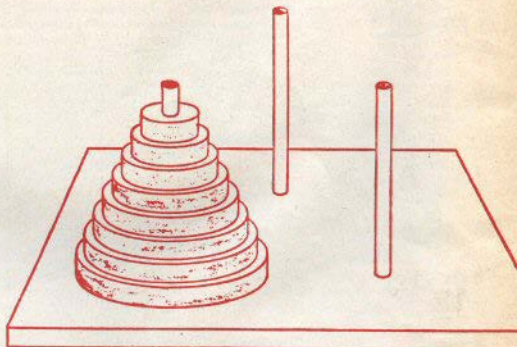
L'illustrazione del metodo seguito per la soluzione "automatica" del problema implica un minimo di conoscenza del concetto di ricorrenza, assente nel basic, e tipica invece di altri linguaggi, particolarmente il PASCAL, a proposito del quale riprenderemo ampiamente il discorso.

Supponiamo di dover spostare N dischi dal piolo A al piolo C, come prescrive il gioco, e supponiamo anche di saperne spostare solo $N-1$: potremmo portare gli $N-1$ dischi dal primo piolo (A) al secondo (B), poi il disco rimanente da A a C; infine sposteremo gli $N-1$ dischi (visto che presumiamo di saperlo fare) da B a C e il problema sarebbe risolto.

Per spostare gli $N-1$ dischi da A a B, cosa che non sappiamo fare, possiamo seguire lo stesso schema, spostando $N-2$ dischi dal piolo di partenza (A) al piolo provvisorio (C) e il disco rimanente al piolo di destinazione (B), per poi spostare nuovamente gli $N-2$ dischi da C alla destinazione B. Visto che non sappiamo spostare nemmeno gli $N-2$ dischi da un piolo di partenza a un piolo di destinazione, ne sposteremo $N-3$ a un piolo provvisorio, uno alla destinazione, e i precedenti $N-3$

La torre di Hanoi

di Fabio Grandi



dalla sede provvisoria alla destinazione: questo schema va ripetuto K volte, finché $N-K=1$, caso in cui sarà uno solo il disco da spostare; questa è l'unica cosa che sappiamo sicuramente fare!

A questo punto, basterà seguire a ritroso la procedura, poiché sapendo spostare 1 disco, potremo spostarne 2, sapendone spostare 2,

ne potremo spostare 3, e così via, fino agli $N-1$ iniziali. Più in generale, lo schema "ricorsivo" da seguire è il seguente: chiamiamo

SPOSTAPILA ($N, part, dest, provv$) la routine che sposta $N-1$ dischi dal piolo *part* al piolo *dest*, attraverso il piolo intermedio *provv*, e MUOVIDISCO (*part, dest*)

la routine che muove un solo di-



**STUDIO D
PER NON SMARRIRE MAI IL FILO DEL DISCORSO.**

STUDIO D

EMITTENTI RADIOTELEVISIVE INDIPENDENTI CHE SI FANNO SENTIRE.

**studio
d**

**CONCESSIONARI MEZZI
RADIOTELEVISIVI**

**STUDIO D
Via Rossini 5 - 20122 MILANO
Tel. (02) 799.592-782.503**



scio dal piolo *part* a quello *dest*; la soluzione sarà data dalla successione di chiamate a subroutine:

```

REM ROUTINE SPOSTAPILA
(N,A,C,B)
IF n=1 THEN PRINT "sposta
da":A;" a ";B"
IF n 1 THEN
SPOSTAPILA (N-1,A,B,C)
(sposta n-1 dischi da A a B usando
il piolo provvisorio C)
MUOVIDISCO (A,C)
(sposta il disco rimanente da A a
C)
SPOSTAPILA (N-1, B,C,A)
(sposta la pila di n-1 dischi da B a
C)

```

Da notare che "spostapila" sarà eseguita ogni volta ricevendo come parametri un numero di dischi da spostare decrementato di uno e le indicazioni su dove vanno spostati i dischi.

Questo implica che va tenuta traccia della strada percorsa, affinché risolto il caso elementare di spostamento di un solo disco si sia in grado di spostarne due, poi tre e così via. Nei linguaggi che imple-

mentano la ricorsione, come il PASCAL, ciò è eseguito duplicando la procedura e i suoi dati *ogni volta* che essa richiama se stessa; così, quando termina l'esecuzione di una procedura chiamata ricorsivamente, cioè dalla procedura stessa, le successive istruzioni eseguite saranno proprie quelle successive alla chiamata della procedura.

La duplicazione e il salvataggio dei dati attuali della procedura al momento della chiamata ricorsiva avvengono su uno STACK, mediante una PUSH di un blocco di valori, rappresentanti appunto i dati. Esisterà, naturalmente, un puntatore all'ultimo blocco salvato su STACK, che consentirà mediante una POP del blocco dei dati di ristabilire lo stato della procedura, prima della chiamata ricorsiva; tale puntatore dovrà quindi essere aggiornato opportunamente dopo ogni operazione di PUSH o POP.

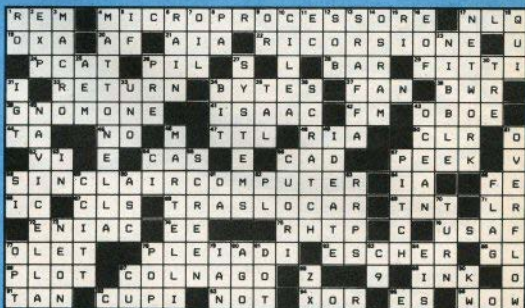
E veniamo finalmente alle implementazioni delle chiamate ricorsive a procedure, che hanno permesso al nostro lettore la soluzione del problema. Per quanto riguarda il

L.M., viene usato lo stack dello Z80 per salvare volta per volta lo stato della routine che chiama se stessa ricorsivamente, attraverso le solite istruzioni assembler PUSH & POP di registri opportuni.

La soluzione basic non è molto differente: viene usato uno stack mascherato, cioè un array o una serie di array (accessibili tutti dalla stessa posizione contemporaneamente, in questo caso) in cui, tramite un contatore con la funzione di stack pointer, vengono effettuate delle assegnazioni all'elemento puntato dal contatore, che viene poi incrementato (simulazione di una PUSH), oppure si assegna ad altre variabili il valore dell'elemento dell'array puntato dal contatore, che sarà poi decrementato (POP): un esempio di questa tecnica è la routine di ordinamento *Quicksort* di F. Berio, pubblicata a pag. 7 del n. 11.

Per quanto riguarda la programmazione strutturata e la ricorsione, vi segnaliamo un titolo fra i tanti disponibili: *Programmazione in PASCAL*, di P. Graogono, ed Muzzio.

Soluzione del cruciverba pubblicato sul n. 15



Implementazione elettronica del gioco d'azzardo per eccellenza, per chi vuole allenarsi senza rischiare soldi nè brutte figure. Le regole (compresa quella che proibisce di guardare le carte altrui) sono quelle classiche del poker, tranne due semplificazioni veniali: la quota d'apertura è fissata casualmente dal computer, e si può accettare anche senza possedere un gioco maggiore della proverbiale coppia di Jack; inoltre di può fare un solo controrilancio: cioè quando l'avversario ha rilanciato e noi avremo accettato, potremo rilanciare a nostra volta per un solo turno.

Il disegno delle carte è forse un po' troppo stilizzato, ma d'altronde il programma è molto compatto. Non sarà difficile, per chi lo vuole, intervenire per migliorare il lato estetico. Agli appassionati di giochi di carte suggeriamo anzi di sviluppare un mazzo completo di carte, con una subroutine autonoma

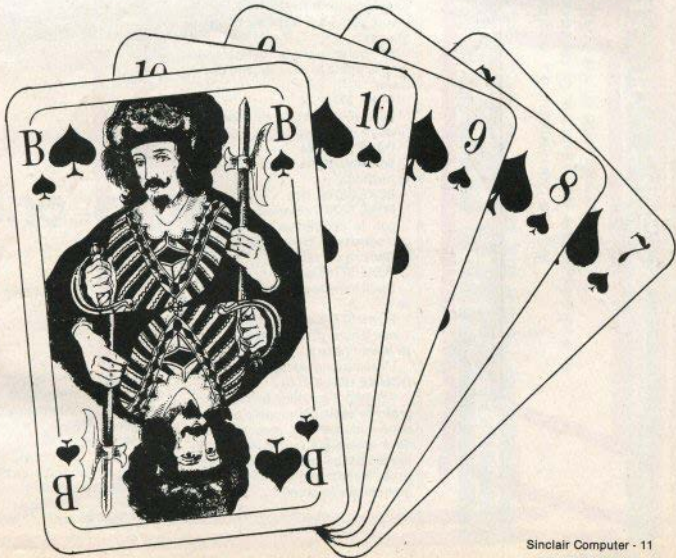
Poker

di G. Beccarini

che possa essere inserita in qualsiasi altro programma di simulazione; chi accogliesse l'invito può naturalmente mandarci il risultato.

L'utente (il giocatore) è sempre guidato sulle operazioni da compiere e non può commettere pratica-

mente errori: il capitale iniziale è di 50.000 dollari (se non vi piace il gioco pesante, consideratele svalutate lirette), e il numero di "smazzate", cioè di partite che si vogliono fare contro lo Spectrum, è stabilito all'inizio del gioco. Buona fortuna.



Window

di Simone Zendrini

Accolto l'invito fatto sul n. 13 a inviare routine per il trasferimento di blocchi dello schermo, l'autore è andato anche oltre le nostre aspettative, con un programma in linguaggio macchina che realizza il trasferimento di finestre di schermo, definite a piacere dall'utente.

Con soli 200 bytes di codici diventa così possibile realizzare ogni sorta di gestione del video a finestre, "quasi" come accade sul fratello maggiore QL. La routine costituisce un ottimo tool (= strumento) con cui lavorare, e ne troverete senza dubbio interessanti applicazioni. Il primo listato è il caricatore dei codici, che esegue anche un controllo sulla correttezza dei dati inseriti; il posizionamento avviene, come di consueto, oltre RAMTOP. A caricamento seguirà il NEW del programma basic.

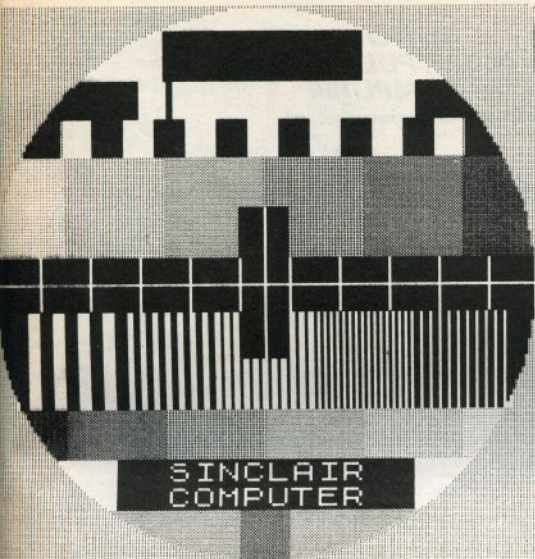
Per poter usare la routine si dovranno effettuare le seguenti operazioni:

- POKE 57782,74
se si vuole portare sul video una finestra già salvata
- POKE 57782,64
per salvare una finestra del video in memoria
- POKE 58260, riga
POKE 58261, colonna
con le coordinate del carattere dell'estremo superiore sinistro della finestra da salvare
- POKE 57766,alt
con il numero di caratteri di altezza e
- POKE 57773, larg
con il num. di caratteri di larghezza della finestra.

L'esecuzione avviene con RANDOMIZE USR 57700.

Il listato 2 contiene un breve programma basic per rendere più semplice l'uso del ling. macchina (che deve essere già in memoria): è numerato dalla linea 9989, per consentire l'inserimento all'interno di programmi già esistenti.





Monoscopio

di C. Padricello

Questo programma disegna sullo schermo un monoscopio. Il disco con la scala cromatica vi sarà utile per la messa a punto dell'immagine del monitor o televisore (linearità, contrasto, centratura, "rotondità" del cerchio...); i fortunati che possiedono un videoregistratore potranno divertirsi a inventare un'emittente "Telespectrum".


RADIOCANESTRO


Tutte le domeniche 32 Radio Locali trasmettono
 le radiocronache delle partite di pallacanestro e Speciale Radiocanestro
 del campionato di A1 e A2 della Lega Basket


CIRCUITO RADIO 84


CONCESSIONARIA PER LA PUGLIA

Via Vespi Scafani 38 - 20146 Milano - Tel. 02/4226065-4224469-4231973-4225876



Speciali confezioni, in formato economico, studiate appositamente per la pulizia dei minicomputers, contengono prodotti, facili da usare, che assicurano una corretta protezione dei videotastiere, drive 3"1/2, 5"1/4. In vendita anche presso i negozi Buffetti.

TUTTO PER LA PULIZIA DEL COMPUTER

La polvere, il fumo, le contaminazioni esterne, possono deteriorare le apparecchiature o cancellare i dati. Un costante uso dei prodotti pulizia, mantiene inalterati dischi, nastri, superfici, carte di credito, ecc. Tutti i prodotti sono omologati dalle migliori case produttrici di drive.

presente alla SMAU
pad. 15/1 stand C 09

DOBELLE



APC

fornisce ogni altro tipo di accessorio per computer.

Distributore esclusivo per l'Italia



00199 Roma, Via Catalani, 23 - Tel. 8392646-8393438 - Telex 621288

La biblioteca degli utenti Sinclair

Mark Harrison

Imparate il Basic con lo ZX81

pagine 132, 13.000 lire

È il microcomputer più venduto al mondo. Ma come ogni computer, necessita di software e documentazione: questo libro contiene 37 programmi che illustrano tutte le caratteristiche e le capacità della macchina.

Robert L. Swarts

ZX80 e ZX81 come strumenti di controllo

pagine 222, 15.000 lire

Questo volume illustra le applicazioni possibili degli home computer nel controllo domestico, fornisce le conoscenze indispensabili e presenta alcuni progetti specifici.

Ray Atherton

Programmare in SuperBasic con il QL

pagine 226, 20.000 lire

Questo libro è una guida per arrivare a una corretta programmazione in SuperBasic. Alla fine di ogni capitolo un questionario e una serie di esercizi di programmazione risolti aiutano il lettore a valutare il proprio livello di apprendimento.

Adrian C. Dickens

L'hardware dello Spectrum

pagine 144, 12.000 lire

Come personalizzare la vostra macchina utilizzando dispositivi elettronici esterni.



muzzio editore

Un regalo della Muzzio per i lettori: a chiunque acquisterà un libro verrà dato in omaggio una copia di "13 giochi con lo ZX Spectrum" di Martin Wren-Hilton il cui prezzo di copertina è di L. 8.000.

Ian Stewart e Robin Jones

Il linguaggio macchina dello Spectrum

pagine 152, 16.000 lire

Un libro per tutti coloro che possiedono un home computer Spectrum e vogliono penetrare più a fondo nelle capacità di questa macchina.

Annibale Izzo

L'ABC del Sinclair ZX80, ZX81, Spectrum

pagine 200, 12.000 lire

Un libro per chi non ha nessuna confidenza con i calcolatori, ma vuole avvicinarsi al mondo dell'informatica domestica e personale e con una guida amichevole e sorridente.

David Laine

ZX Spectrum: tecniche avanzate di linguaggio macchina

pagine 176, 18.000 lire

Consente agli esperti un uso più sofisticato del computer. Vi sono descritte routine particolarmente complesse e ognuna di queste è illustrata con disegni e diagrammi.

Mark Harrison

Imparate il Basic con lo Spectrum

pagine 192, 19.000 lire

Un libro per chi non ha esperienza di computer, e per chi è già esperto ma necessita di maggiori informazioni di quelle del manuale. Partendo dai principi di base, il lettore può arrivare alle più avanzate tecniche utilizzabili.

Desidero acquistare

-

Pagherò al postino il prezzo indicato + L. 1.000 per contributo spese postali

Desidero ricevere solo il Vs. catalogo generale

Ritagliate e spedite a:
gruppo editoriale muzzio

via marziale 73, 35138 padova

nome cognome

via città

C.a.p.





SPAZIO QL

COMPACT numero canale (compreso fra 0 e 2) stringe la spaziatura orizzontale e verticale della finestra specificata, che di conseguenza aumenta il numero di caratteri visualizzabili.

```
100 DEFine PROCEDURE compact(n)
110 IF n>2 THEN RETURN
120 CSIZE#n,0,0
130 POKE_W 167424+256*n+30,6
140 POKE_W 167424+256*n+40,9
150 END DEFine compact
10000 OPEN#3,seri
10010 BAUD 9600
10015 PRINT#3,CHR$(27);"M";
10020 PRINT#3,CHR$(27);"A";CHR$(10);C
HR$(27);"B";CHR$(27);"I";CHR$(10);CHR
$(27);"0";CHR$(47);
10030 LIST#3
```

Creare un file b\$ uguale alla sorgente a\$, ma dotato di autostart: gli effetti sono gli stessi ottenibili sullo Spectrum, solo che sul QL un programma basic con autostart parte automaticamente, anche se si usa MERGE per tentare di aggirare la protezione. Esempio d'uso: AUTOSTART "mdv1 pippo", "mdv1 pippo 2"

```
100 DEFine PROCEDURE autostart(a$,b$)
110 DELETE b$
120 OPEN#197,a$
130 OPEN_NEW#198,b$.
140 REPEAT ciclo
150 IF NOT EOF(#197)THEN
160 PRINT#198,INKEY$(#197,-1);
170 ELSE
180 EXIT ciclo
190 END IF
200 END REPEAT ciclo
210 CLOSE#197;CLOSE#198
220 END DEFine autostart
```

Creare due copie di sicurezza di un file, nella stessa cartuccia in cui esso già si trova, eliminando prima le copie preesistenti.

Utility

```
100 DEFine PROCEDURE sec_copy(a$)
110 DELETE a$;"_sec"
120 DELETE a$;"_sec_sec"
130 COPY a$ TO a$;"_sec"
140 COPY a$ TO a$;"_sec_sec"
150 END DEFine sec_copy
100 DEFine PROCEDURE curs_speed(d,f)
110 POKE_W 163980,d
120 POKE_W 163982,f
130 END DEFine curs_speed
```

Rende possibile la manipolazione dei valori riguardanti il ritardo d'attivazione e la frequenza dell'autorepeat dei tasti: le variabili di sistema 163980 w e 163982 w equivalgono alle spectrumiane 23562 e 23561, ma i valori POKEati hanno effetti diversi.

```
100 DEFine PROCEDURE curs_speed(d,f)
110 POKE_W 163980,d
120 POKE_W 163982,f
130 END DEFine curs_speed
```

Una procedura per giochi che sfrutta la possibilità di disattivare la lettura del display-file da parte dell'ULA (con POKE 98403,0) e di ripristinarla subito dopo (POKE 98403,128). L'effetto ottenibile è interessante, e può essere leggermente modificato inserendo delle pause tra le due POKE. La durata è il numero di ripetizioni richieste: per poter apprezzare gli effetti ottenuti si consiglia di usare valori piuttosto alti (es. 1000).

```
100 DEFine PROCEDURE crash(durata)
110 LOCAL r
120 FOR r=1 TO durata
130 POKE 98403,0
140 POKE 98403,128
150 END FOR r
160 END DEFine crash
```

Posta

Il QL e lo Spectrum

È vero che il QL disporrà di un adattatore per far girare i programmi dello Spectrum? Vorrei un programma traduttore per trasferire i programmi Spectrum al QL. (vari lettori)

Queste e simili domande giungono con frequenza in redazione: le voci su un fantomatico "adattatore" circolano già dall'uscita del QL, ma di concreto non abbiamo ancora visto niente. Teoricamente, la cosa è fattibile (esiste già un interprete del basic Spectrum per il C64), ma non certo facile: è probabile che qualcuno, nel segreto di una software-house, ci stia lavorando; una soluzione hardware non ci sembra economicamente praticabile.

Vocali accentate

Semberebbe che dal word-processor QUILL non siano utilizzabili i caratteri del superbasic raglungibili con CTRL e SHIFT (quali, per esempio, le vocali accentate).

Il difetto, lamentato da alcuni utenti, è effettivamente riscontrabile nella versione 1.00 del QUILL (la sigla relativa alla versione appare sullo schermo all'inizio del caricamento). Non è più presente nella nuova versione, che accetta tutti i caratteri disponibili da basic, comprese quindi le vocali accentate.

Una nota informativa della Sinclair comunica che i possessori delle vecchie versioni di tutti i 4 package, QUILL, EASEL, ARCHIVE, ABACUS, possono ricevere le nuove gratuitamente, inviando le cartucce originali (di cui conservate ovviamente i back-up!). Poiché il metodo, anche se esente da rischi, non è certo veloce, forse vi conviene contattare un neo-acquirente, nella vostra zona o tramite SINCLAIR COMPUTER, per avere da lui una copia.

Fermare lo SCROLL

Apparentemente il QL non possiede un comando che sospenda lo scrolling di una finestra video: sarebbe una lacuna fastidiosa, non poter esaminare un listato pagina per pagina o non riuscire a leggere una directory più lunga di 20 titoli.

In realtà la carenza è solo nelle istruzioni, basta premere CTRL+ F5, e lo scroll si arresta; riprende con lo stesso comando, o con qualsiasi altro tasto (che però viene bufferizzato, ed eseguito al termine dello scrolling, mentre F5 non ha effetti collaterali). Funziona anche come BREAK temporaneo di un programma basic.

Microdrives random

Il sistema di accesso casuale ai microdrives dello Spectrum, pubblicato su SC n. 12, è utilizzabile anche sul QL? (E. Marinacci - Roma)

Non è necessario: il QDOS dispone già di routines proprie per una gestione ottimizzata delle cartucce, con un sistema di posizionamento dei puntatori, all'interno dei files di dati, per eseguire la lettura random in modo molto efficiente (considerata la struttura fisicamente sequenziale del supporto); e infatti il tempo medio di accesso, poco più di 3", è sensibilmente inferiore a quello dei microdrives dello Spectrum.

Versione italiana

Quali differenze vi sono tra la versione inglese e quella italiana? (C. Frigeni-Bonate BG) La più volte annunciata versione italiana, ai pari di quella francese o tedesca, non è ancora giunta in mano a comuni mortali, né sappiamo dirvi quando arriverà: le date dei comunicati stampa lasciano troppo spesso il tempo (e la tastiera) che trovano. Personalmente, non ne sentiamo molto la mancanza: i tasti "all'italiana", per programmare, danno più problemi che vantaggi.

La modifica più appariscente sarà comunque proprio la tastiera, poiché l'hardware dovrà mantenere, necessariamente, la totale compatibilità. Non è parimenti ancora chiaro se e quando saranno disponibili in italiano i 4 package di sup-

porto.

QL & stampanti

Si ripetono con il QL le domande relative all'interfacciamento con le stampanti più disparate, comprese le macchine per scrivere elettroniche (tipo Olivetti serie ET), e con i monitor. Ripetiamo perciò alcune considerazioni di carattere generale.

a) potete utilizzare qualsiasi stampante dotata di porta seriale (RS-232C) o di porta parallela (Centronics); nel primo caso il collegamento è diretto, tramite la porta SER1 del computer, nel secondo avete bisogno di un convertitore seriale/parallelo, che il rivenditore vi deve fornire unitamente alla stampante e che si inserisce sempre in SER1. Si tratta di uno scatlino contenente un circuito piuttosto elementare: purtroppo l'abbiamo visto in vendita anche a 170.000 lire (al limite del furto).

b) il pilotaggio della stampante avviene nei modi consueti, con l'invio di caratteri di controllo; se è Epson-compatibile, il QUILL è già predisposto con i giusti codici, diversamente si rende indispensabile consultare congiuntamente i manuali della stampante e del QUILL stesso.

c) Chi desidera l'alta qualità è inevitabilmente attratto dalle stampanti a margherita. A favore hanno la stampa praticamente perfetta, contro il prezzo superiore, la lentezza, il set di caratteri limitato e non programmabile, l'impossibilità di hard-copy e di qualsiasi opzione grafica; è inoltre opportuno verificare l'interfacciabilità prima dell'acquisto.

d) Quanto ai monitor, il connettore DIN del QL ha i segnali adatti sia per gli ingressi PAL-CCIR (oVCR) che RGB; nel capitolo Concepts del manuale trovate la piedinatura. Manca invece il segnale audio, che può essere recuperato, come nello Spectrum, presso l'altoparlantino interno.

L'ultima versione

Curiosità e incertezza anche riguardo a nuove versioni della ROM: le riviste inglesi hanno scritto, con



SPAZIO QL

Il solito eccessivo anticipo, di una nuova ROM, che sostituisce la versione JM (verificabile con PRINT VERS), e denominata JS. Ora esiste realmente, possiede nuovi comandi, sono state effettuate correzioni a piccole "incertezze" della JM, e NON sarà distribuita gratuitamente ai possessori della serie precedente. Non cercatela nel QL dei negozi italiani: per ora è distribuita in Inghilterra.

Salvataggio DATA

Come si può salvare un array sulla cartidge? Dove si trovano le variabili di sistema? (L. Zammarchi - Follonica GR)

La possibilità di salvare un array, possibile sullo Spectrum con SAVE "nome" DATA, è utile con il registratore, che non consente la gestione "aperta" di un file di dati. Si può ottenere qualcosa di simile ponendo i numeri o le stringhe in linee DATA e salvando solo questa parte del programma basic.

SAVE...nome, (num. linea) TO (num. linea).

o aprendo un canale verso un file sul nastro (con OPEN), e salvandovi in sequenza i dati (un ciclo di PRINT sul canale).

Ripareremo ampiamente della gestione dei files, così come delle variabili di sistema, che abbiamo iniziato a esaminare sul n. 14.

Come si programmano da basic i tasti funzione? Come funziona il comando SEXEC mdvn...? (M. Baragona - Roma)

I tasti funzione da basic non sono utilizzabili: forse lo saranno nelle future versioni.

SEXEC serve per salvare sulla cartuccia un programma in linguaggio macchina, che potrà in seguito essere richiamato ed eseguito in multitasking con EXEC o EXEC W, dove W significa WAIT: il basic viene sospeso fino alla fine del L/M.

I parametri che richiede, dopo il numero del drive e il nome del file

con cui salvare, sono l'indirizzo di inizio del programma (dove lo si è scritto), la lunghezza totale dell'area riservata e la quantità di memoria richiesta dal programma stesso per memorizzare dati. L'uso di questi comandi richiede però una buona conoscenza del QDOS, che non può essere improvvisata.

I modi di EDIT

Nel libro "Alla scoperta del QL", di A. Nelson, edito anche in italiano, vi sono spesso imprecisioni che possono generare confusione; molto probabilmente è stato scritto prima dell'uscita della versione JM della ROM. In un caso l'imprecisione diventa un vero e proprio errore: "...con EDIT si può lavorare in due modi: inserimento e modifica (con sovrapposizione). Per passare da una modalità all'altra basta premere il tasto ALT." (pag. 27)

Questa caratteristica non esiste affatto nel QL attualmente circolanti; è presente invece nel toolkit su cartidge, della Sinclair, che implementa anche molte altre ottime utilità.

Super basic

di Marco Mussini

Un po' di teoria

L'input/output consiste nella comunicazione che il computer intavola in continuazione con periferiche e dispositivi collegati alla CPU; tale comunicazione può svolgersi in uno o due sensi, un bit o un byte per volta: il formato e la direzione del flusso (STREAM) di dati sono determinati dal tipo di periferica (DEVICE) di cui si sta facendo uso. Poiché il QL può essere collegato contemporaneamente a più di un device, il QDOS deve poterli distinguere gli uni dagli altri (potremmo avere collegate per esempio 2 stampanti)

TIRRADIO
KHF

Il più grande Network Italiano

100 RADIO

6 ore di programmi giornalieri simultanei

CIRCUITO RADIO 84

CONCESSIONARIA PER LA PUBBLICITÀ

Via Vesprì Siciliani 38 - 20146 Milano - Tel. 02/4226065-4224469-4231923-4225876



e soprattutto disporre di un'area di controllo (CHANNEL DEFINITION BLOCK), a uso esclusivo delle routines che li pilotano (device drivers); da qui la necessità di potersi riferire a una periferica solo specificandone il nome, o meglio un codice numerico (in L/M è a 32 bit e si chiama CHANNEL ID, in BASIC è un numero intero compreso fra 0 e 255, quindi a 8 bit), che si riferisca a esso in modo univoco, per semplificare la distinzione fra i vari canali.

I canali e i device drivers

Avrete capito che il QL non «vede» direttamente le periferiche che gli sono state collegate, ma solo dei canali logici ai quali si rivolge per mandare o ricevere dati. Questi canali fanno uso dei DEVICE DRIVERS (cioè delle routines in linguaggio macchina che eseguono fisicamente la trasmissione o la ricezione dei dati dai devices, ovvero pilotano i devices stessi; di qui il nome di queste routines), per portare a termine l'effettivo dialogo con le periferiche; al QDOS non interessa, per esempio, come fa l'hardware a spedire un dato alla porta seriale o a riceverlo dalla testina del microdrive.

Tutte queste informazioni devono essere note alle routines che gestiscono fisicamente le periferiche (i device drivers, appunto); al QDOS basta invece sapere che, per mandare un dato a un device, deve inviarsi al canale logico a esso collegato o, meglio, assegnato dall'utente.

Ne discende che, per il QL, i canali si gestiscono tutti allo stesso modo, tant'è vero che si comunica con una porta seriale così come si comunica con una finestra video, mentre le periferiche sono trattate ognuna a modo suo dai relativi device drivers, ed è proprio per questo che il QDOS è definito *peripheral in-*

dependent operating system.

Un altro vantaggio di questa impostazione è che al repertorio (LINKED LIST) di device drivers incorporati nel QDOS, se ne possono aggiungere alcuni altri, scritti dall'utente, anch'essi in codice macchina, adatti alla periferica che si vuole interfacciare; per esempio «flip» per un floppy o «par» per una Centronics. Per questo si dice anche che il QDOS è *modulare* ed *estensibile*, mediante la semplice aggiunta di nuovi moduli.

Prima di addentrarci nella trattazione dei singoli devices, è opportuno analizzare le procedure da seguire per aprire, chiudere e usare un canale.

La gestione dei canali in superbasic

Per aprire un canale è necessario comunicare al computer che d'ora in poi tutte le operazioni di I/O rivolte verso questo canale dovranno intendersi riferite al device cui esso sarà stato collegato. L'istruzione che il Super Basic mette a disposizione per comunicare al QDOS questa serie d'informazioni è OPEN, con la sintassi:

OPEN # numero canale, nome device

Per esempio, facendo eseguire
OPEN # 35,ser2

si apre un canale (il n. 35) collegato alla porta seriale 2; da questo momento, fino a quando il canale 35 verrà chiuso, cioè disattivato, ogni comando di I/O rivolto a questo canale avrà l'effetto di mandare o ricevere dati attraverso tale porta.

Per chiudere un canale si usa invece l'istruzione CLOSE, la cui sintassi è

CLOSE # numero canale

e il cui effetto è quello di svuotare tutti i buffers eventualmente associati al canale specificato (situazione che si verifica sempre con i Microdrives), di disattivare le comunicazioni con la periferica cui esso era collegato, e di lasciare libera l'area di memoria fino a quel momento usata come CHANNEL DEFINITION BLOCK del canale. Per esempio, con

CLOSE # 0

si esclude la tastiera dalla finestra comandi, inceppando il calcolatore (come digitare sullo Spectrum POKE 23659,0); questo trucco può fungere da sistema di protezione anti-BREAK, ma andateci piano: il computer si implanta ogni volta che nel canale 0 debba venir visualizzato qualcosa, cursore o messaggio d'errore che sia. Il programma che proteggerete con questo comando dovrà quindi essere assolutamente «bug-free» e non usare mai la finestra 0. In caso contrario, ... tiiti.

Parliamo con le periferiche

Il dialogo con le periferiche si svolge, ovviamente, mandando o ricevendo dei dati. Ci sono due comandi e due funzioni adibiti a questo scopo:

PRINT, INPUT, INKEY\$, EOF

PRINT [# canale,] [espressioni da inviare, divise da separatori]

Invia nel canale specificando (altrimenti, per default, nel canale 1) l'espressione numerica o stringa sotto forma di caratteri ASCII. Vale a dire che se vogliamo mandare alla stampante il carattere di controllo 22, non dovremo scrivere «22», né 22: in entrambi i casi troveremo scritto il numero 22. Per mandare un byte di valore ben preciso occorre usare la funzione CHR\$, in modo da creare una stringa costituita dal carattere il cui codice ASCII è quello desiderato.

In pratica, l'argomento della PRINT deve essere per forza una stringa; se è una variabile numerica o un numero, il computer lo converte in una stringa per mezzo della *coercion*.

Altro punto da ricordare è la distinzione tra i possibili separatori; nello schema di sintassi abbiamo usato la virgola, ma avrebbe anche potuto essere un punto e virgola, una barra (slash) rovesciata o un punto esclamativo. La distinzione reale fra questi separatori si sente solo quando essi sono usati sullo schermo (se ne parlerà più oltre), per ora vi basti ricordare che con qualsiasi periferica il punto e virgola non ha alcun effetto (funge cioè



solo da elemento sintattico), mentre la barra rovesciata esegue un CR (Carriage Return) e un LF (Line Feed) (CHR\$(13 e 10): in pratica va a capo.

Con le porte seriali si può scegliere se includere o meno il CR; con lo schermo esso non viene mai inviato, perché il device driver dello schermo esegue da solo un CR quando riceve un LF. Inviando un CHR\$(13) allo schermo si otterrà un simbolo a scacchiera che indica che il carattere non è stampabile e non è stato preso in considerazione dal device driver.

L'uso dei caratteri il cui codice è inferiore a 32 o superiore a 191 è soggetto a variazioni, in base al dispositivo al quale vengono inviati.

Un separatore può trovarsi anche alla fine di ogni espressione numerica o stringa, per decidere dove dovrà finire il successivo output rivolto a quel canale. In mancanza di separatore, il QL manda un «a capo», altrimenti si regola in funzione del separatore usato nel comando.

Nel caso che si vogliono accostare varie espressioni, esse dovranno essere delimitate da un separatore.

Per ricevere un carattere da un canale si usa invece la funzione

INKEY\$(canale, tempo d'attesa dati)

Questa istruzione, nella sua forma più semplice, serve per rilevare pressioni di tasto, ma nel QL essa viene usata anche per prelevare by-

tes da un qualunque dispositivo, attraverso un canale aperto verso di esso. Il canale deve essere specificato, ovviamente, per far capire al computer dove dovranno essere pescati questi bytes.

Il tempo d'attesa dati comunica al computer che il dato dovrà essere atteso per un tempo pari a n millisecondi (n è il valore specificato). In pratica diciamo al computer di recarsi all'appuntamento con i bytes e di attenderli, nel caso dovessero tardare, per un certo periodo di tempo, dopodiché dovrà... tornare a casa. Se si vuole che la pazienza del computer sia illimitata, basta specificare -1 come tempo d'attesa.

Se tale tempo non viene espresso, il computer assume 0: in questo caso, se il dato è già presente nel buffer (il luogo dell'appuntamento), viene prelevato; in caso contrario, il computer non lo aspetta e ritorna subito indietro.

INKEY\$ ritorna quindi il primo carattere in arrivo dal canale specificato; se tuttavia in questo canale non ci sono dati o non sono ancora arrivati dal device, viene ritornata una stringa nulla ("").

Quando si deve caricare da un canale non un carattere ma un'intera stringa (max 127 caratteri) o un valore numerico, si può usare il comando INPUT, che funzionalmente è il contrario di PRINT, anche se ci sono due differenze sostanziali: poiché il buffer utilizzato per trasferire i dati è lungo 128 bytes, INPUT può trattare stringhe di lunghezza non superiori appunto a 127 caratteri (+1 ENTER = 128 bytes), a differenza di PRINT, che non ha limitazioni per quanto riguarda la mole dei dati da trasferire; inoltre mentre PRINT può mandare soltanto dei bytes, soli o raggruppati in stringhe, INPUT può interpretare i dati di ingresso come stringa, come variabile floating point o come variabile intera, a seconda del tipo della variabile destinata ad accoglierli.

INPUT può provocare un messaggio «BUFFER FULL» nel caso che il QDOS abbia saturato il buffer senza incontrare nemmeno un CHR\$(10) che delimitasse il blocco di dati da ricevere dal canale. Oppure può capitare un messaggio «ERROR IN EXPRESSION», qualora i

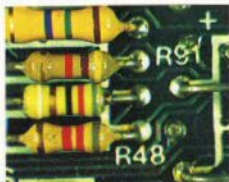
dati inviati dal device non possano essere convertiti nel formato adatto con la coercion. Infine potremo incontrare un messaggio d'errore «END OF FILE», che comunica all'operatore che nel canale non ci sono più dati. Nel caso di un canale aperto su Microdrive significa che il file è stato letto fino alla fine e che di conseguenza non se ne può cavare alcun altro dato. Per sapere in anticipo che il file è terminato, prima cioè di incorrere nella condizione d'errore, esiste la funzione EOF (canale), che ritorna 1 se il file è terminato, o 0 se ci sono ancora dei dati da leggere. Per inciso, la stessa funzione, senza parentesi e senza indicazione del canale, comunica se ci sono ancora o no dei DATA da leggere in un programma BASIC. I due possibili valori ritornati da EOF sono organizzati in modo tale da permetterne nel modo più semplice possibile l'uso in comandi IF: per esempio si potrebbe scrivere facilmente un programma per visualizzare tutti i dati ricevuti da un canale a scelta, in grado di prevenire l'errore END OF FILE:

```
100 INPUT n
110 REPEAT view
120 IF EOF(n) THEN
130 PRINT "Il canale 'n'; non ha più
dati";EXIT view
140 ELSE
150 PRINT INKEY$(n,-1);BEEP 10,10
160 END IF
170 END REPEAT view
```

EOF agisce come una funzione booleana: vale 1 se è vera e 0 se è falsa. Questo, come abbiamo visto, per facilitarne l'inserimento negli IF.THEN.

Conclusione

Per questo mese ci fermiamo qui; nella seconda parte si parlerà dei devices disponibili nel QL, analizzandoli singolarmente, senza tralasciare le informazioni di cui ha bisogno chi programma in linguaggio macchina.



Introduzione al



DATA
BASE

di Mariangela Guardione

Un po' di teoria del data base. Per la pratica, cercate nelle edicole la nostra cassetta.

Lo sviluppo sempre più rapido all'interno dell'ambito di lavoro dei terminali ha notevolmente accresciuto, in questi anni, l'importanza dell'organizzazione del data-base, che rappresenta un archivio, in generale centralizzato, su memorie ad accesso diretto, contenente tutti i dati relativi ad una periferia normalmente vasta di utenti che devono accedere a queste informazioni.

In un contesto sempre più informatizzato il data-base ed il continuo evol-

versi delle relative tecnologie forniscono sia all'analista che al programmatore la possibilità di sfruttare completamente la capacità di elaborazione delle informazioni di un elaboratore digitale. Infatti l'unico scopo di un data-base è quello di facilitare la generazione di informazioni e quindi dovrebbe essere utilizzato come struttura di supporto per le informazioni. Tutto questo è importante in quanto l'analisi e l'organizzazione logica dei dati sono parti

essenziali del processo da cui deriva la conoscenza.

Una delle caratteristiche fondamentali del data-base è quella di essere abbastanza vasto da permettere non solo di gestire l'elaborazione delle informazioni in esso contenute, ma anche di mantenere nel tempo sia la validità che l'affidabilità delle stesse. In altri termini questa struttura deve essere sufficientemente estesa da supportare gli innumerevoli problemi che



nascono dalla necessità di effettuare modifiche o variazioni ed è questo il motivo per cui un sistema di elaborazione deve essere fondato su solide basi.

Il termine dominante in ogni ambito produttivo e su cui si basa lo sviluppo sia tecnico che economico di una industria è **informazione**.

Questa parola indica la conoscenza derivante dall'elaborazione del dato che rappresenta un elemento effettivo o supposto, utilizzato poi come base per un calcolo. I dati relativi ad una azienda e alle sue relazioni con la società nella quale agisce, sono indipendenti dal management aziendale, mentre solo le informazioni che possono essere generate da un data-base riflettono le caratteristiche tipiche del management.

Si è definito precedentemente un dato come elemento e questo porta ad identificarlo ed a descriverlo; tutto questo implica che un data-base non sia limitato ai soli fatti, ma possa comprendere:

- previsioni;
- ipotesi non verificate;
- supposizioni.

Inoltre un'altra caratteristica del dato è quella di poter essere "utilizzato" ed è questa la problematica più difficile da affrontare in fase di progettazione di un data-base. Un utilizzo errato dei dati rischia di compromettere le caratteristiche di affidabilità e validità nel tempo del data-base stesso, in quanto il dato rappresenta un elemento base per il calcolo e quindi deve fornire il valore quantitativo usato per arrivare alla conoscenza di un fatto, per legittimare l'affermazione che il contenuto di un data-base deve essere utile per la produzione di informazioni.

Il tipo e la quantità dei dati devono essere adeguati alle esigenze di conoscenza dell'azienda, in quanto la principale differenza esistente fra le caratteristiche "utilizzato" e "base" per un calcolo è analoga a quella tra uso potenziale ed effettivo del dato. Infatti la forma si riferisce da un punto di vista logico ai dati che potrebbero essere contenuti in un data-base, mentre la seconda concerne il contenuto attuale di informazioni da un punto di vista fisico o quantitativo.

Le organizzazioni tradizionali dei dati sono basate, da un punto di vista logico, su due concetti fondamentali:

- un insieme di dati da elaborare è organizzato in files e questi sono strutturati al loro interno in blocchi fisici (records e campi);
- la ricerca dei dati in un file avviene

assegnando una chiave di ricerca del file, formata da uno o più campi, che identificano univocamente un record.

I files di dati sono sempre strutturati in funzione dei programmi che li devono utilizzare, per cui i problemi tecnici di programmazione vengono ad avere un ruolo predominante rispetto a quelli di organizzazione dei dati.

In ogni programma è quindi necessario specificare esattamente i files che si intendono utilizzare, l'organizzazione, l'accesso e le caratteristiche fisiche di memorizzazione.

Tutte queste considerazioni pongono in evidenza una serie di svantaggi che emergono dall'organizzazione tradizionale dei dati in strutture non integrate:

● **dipendenza dati-programmi:** al variare delle specifiche, in un programma può cambiare la struttura dei dati che devono essere manipolati;

● **molteplicità dei files:** ogni procedura ne richiede di specifici, in quanto molto spesso, al crescere delle procedure applicative, aumenta notevolmente anche il numero dei files;

● **ridondanza dei dati:** la pluralità di files comporta che spesso informazioni uguali vengano memorizzate su files diversi, con inutile occupazione di memoria;

● **molteplicità degli aggiornamenti:** per rendere un'informazione inconsistente è necessario aggiornare tutti i files contenenti quella specifica informazione simultaneamente;

● **dipendenza dati-sistema:** spesso si rende necessario fondere più files per esigenze di configurazione del sistema.

Infatti in queste situazioni un insieme di files che memorizza informazioni riguardanti una determinata realtà ha il principale inconveniente che i diversi files esistono come entità distinte che vengono gestite come un tutto unico tramite programmi applicativi. Le esigenze reali richiedono normalmente più copie delle stesse informazioni organizzate in maniere diverse.

Tutto questo porta ai moltiplicarsi di supporti fisici di memorizzazione e di accessi alla memoria secondaria; tali inconvenienti aumentano quanto più cresce la complessità del sistema di informazione che si vuole realizzare.

Come esempio di tutto questo si pensi alla gestione di una biblioteca, in cui le informazioni non siano statiche, ma possano variare nel tempo: nuovi libri da inserire in biblioteca, libri persi da cancellare, informazioni da modificare riguardanti alcuni libri.

L'operazione di inserimento di un

nuovo libro comporta come conseguenza l'inserimento di nuovi dati in tutti i files nel momento in cui si conoscono tutte le informazioni relative, quali per esempio la sua collocazione in biblioteca. Ma anche l'operazione di cancellazione deve essere eseguita contemporaneamente su tutti i files allo scopo di non mantenere informazioni inconsistenti. A ciò si aggiungono anche le modifiche ad un'informazione, come l'acquisto di un libro già presente in biblioteca, ma che presenta una data di edizione più recente che implica la necessità di un aggiornamento degli stessi files. La causa di questi molteplici aggiornamenti risiede nella sovrabbondanza delle informazioni presenti nei files implicati nella gestione della struttura che si vuol esaminare.

Per evitare queste difficoltà è necessario adottare delle tecniche di gestione più sofisticate e più razionali, quali ad esempio il data-base.

L'architettura

Questa struttura operativa può essere definita nella seguente maniera: un data-base è un insieme di dati che deve essere utilizzato da tutti gli utenti e dai programmi applicativi di un sistema informativo; è costituito da sottosistemi scollegati fra loro, e possiede una rappresentazione logica dei dati indipendente dalla effettiva memorizzazione fisica di questi.

Tutti questi requisiti sono determinanti per formalizzare l'architettura di un sistema di gestione. La definizione fornita, basata su concetti di multiutenza, integrazione ed indipendenza dalla rappresentazione dei dati, conferisce ai sistemi che operano in ambiente data-base una serie di vantaggi che sono:

● **indipendenza della struttura fisica:** eventuali modifiche come l'aggiunta di nuovi tipi di dati e variazioni al tipo di organizzazione o accesso ai dati non devono variare i programmi che operano sul data-base;

● **indipendenza della struttura logica:** deve essere fornita la possibilità di definire delle "viste utente" che sono del tutto indipendenti dallo schema generale del data-base;

● **non ridondanza dei dati:** sono in genere presenti una sola volta dati di informazione relative a più applicazioni;

● **possibilità di rappresentare**

relazioni comunque complesse fra i dati: una ben precisa situazione reale è assai complessa, ed esistono fra i dati dalle relazioni che non possono essere schematizzate semplicemente;

● **sicurezza dei dati:** un data-base, essendo un archivio integrato, contiene dati che sono utilizzati da più utenti e quindi deve possedere delle procedure che consentano di proteggere le informazioni sia da accessi non autorizzati che da guasti hardware e/o software e assicurare l'esatto contenuto di un dato per evitare manipolazioni non autorizzate.

Data-base e procedure

Un data-base viene gestito da un insieme di procedure, che permettono la creazione della struttura dei dati ed il controllo e l'accesso a questi, che vengono indicate con il termine DBMS, derivante da Data Base Management System.

La struttura in oggetto è quella che permette di definire a livello logico l'organizzazione dei dati, controllare a livello fisico la loro memorizzazione e le procedure che consentono la sicurezza degli stessi, accedere ai dati dalla loro definizione logica e permettere un'alta indipendenza tra quella che rappresenta la struttura logica e la memorizzazione fisica dei dati. Per ottenere tutto questo, i DBMS oggi esistenti sul mercato sono strutturati utilizzando dei livelli di astrazione. Vi sono diversi schemi che rappresentano i dati e che si riferiscono ciascuno a un ben determinato livello. E' importante osservare che per successivi livelli di astrazione possono esistere:

● **uno schema fisico dei dati** che permette di definire sia le aree di occupazione che i criteri di accesso alle informazioni;

● **uno schema logico dei dati** che determina sia le aggregazioni che le dipendenze che si possono creare nei dati, la cui memorizzazione viene ottenuta mediante l'impiego di un modello di dati;

● **uno o più schemi esterni**, che creano le viste utente, cioè quelle porzioni dello schema logico generale del livello sottostante, visibili all'operatore.

Un'altra funzione del DBMS è quella di colloquiare con il sistema operativo allo scopo di poter accedere fisicamente alla memoria secondaria e di

gestire quindi l'I/O. Quest'ultima operazione, che consente un dialogo con l'esterno tramite i programmi applicativi e le interfacce in tempo reale, viene ottenuta con l'impiego del livello di astrazione superiore. Scopo dell'utilizzo di un sistema di calcolo automatico è quello di simulare la realtà che si presenta sempre in strutture complesse, in quanto esistono delle entità che si possono presentare come insieme di dati elementari che sono però tra loro in relazione non semplice. Per questo motivo è utile adottare dei criteri di formalizzazione che introducono l'impiego di un modello che, se sviluppato in maniera corretta, descrive in modo adeguato la realtà che si vuole gestire. L'utilizzo dei modelli è una fase molto importante che precede l'effettiva realizzazione di un'opera ed è questo il motivo per cui è indispensabile conoscere quali sono i modelli che vengono tradizionalmente utilizzati in questa delicata fase di realizzazione di un progetto. Essi possono essere sostanzialmente tre:

- modello gerarchico;
- modello reticolare;
- modello relazionale.

Dipendenze

Si è parlato precedentemente delle relazioni che possono esistere tra due insiemi di dati e che possono essere descritte, supponendo che A e B siano i due insiemi, nel seguente modo:

● **dipendenza 1:1;** questo significa che ogni elemento che appartiene ad A è legato al più con un elemento di B e viceversa; per esempio:

DIPENDENZA 1:1

Se in una biblioteca non si hanno autori omonimi si verifica la dipendenza:

NOME AUTORE <--> DATA NASCITA AUTORE

in quanto ogni autore è nato in un determinato giorno/mese/anno. Questa dipendenza è rappresentata in figura 1;

● **dipendenza 1:N;** in cui ogni elemento di A è in relazione con uno o più elementi di B. In senso opposto significa che ogni elemento di B è legato a un solo elemento di A; per esempio:

DIPENDENZA 1:N

LIBRO <--> COLLOCAZIONE

Infatti ogni libro è in generale presente nella biblioteca più volte: questo indica che ha più collocazioni associate. Al contrario, fissando una certa collocazione, si ottiene uno e un solo libro associato. La rappresentazione grafica di questa dipendenza è data dall'immagine 2.

● **dipendenza M:N;** in cui ogni elemento di A è legato con uno o più elementi di B e viceversa; per esempio:

DIPENDENZA M:N

AUTORE <-->> LIBRO

Questo esempio spiega che ogni libro è scritto da uno o più autori e ciascun autore firma uno o più libri (fig.2a).

A questo punto passiamo ad esaminare i tre tipi di modelli per rappresentare la realtà partendo da quello che

FIG. 1

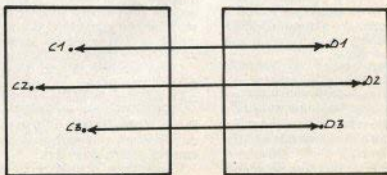


FIG. 2

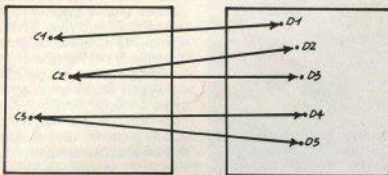
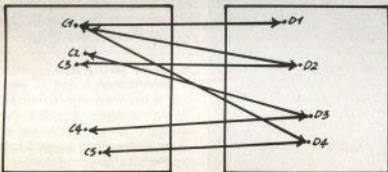


FIG. 2A



storicamente è stato il primo: il Modello Gerarchico.

L'albero

In questa rappresentazione le informazioni sono organizzate secondo una struttura ad albero così costituita:

- radice dell'albero;
- padri e figli;
- livelli dell'albero;
- possibili legami padre-figlio e viceversa.

Questa struttura gerarchica crea l'esistenza di un nodo radice, il cui scopo è quello di collegare tutte le informazioni facenti parte di un determinato albero, che poi si dirama in un insieme di padri, figli, figli dei figli; risulta perciò possibile definire le parentele che esistono fra i vari componenti dell'albero.

I suoi livelli possono essere definiti come le generazioni presenti nell'albero. Tutto questo implica la presenza di vincoli sui possibili legami fra le varie informazioni presenti nella struttura, rappresentanti la stessa, permettendo a quest'ultima di evolversi unicamente in determinati modi.

(1 - continua)

Trasferimenti al microdrive

a cura di Marco Bertani

Uno dei vantaggi fondamentali dei microdrives rispetto al normale registratore è evidentemente la velocità: in pochi secondi è possibile salvare o caricare in RAM blocchi di bytes lunghi quasi quanto la RAM stessa, che siano essi programmi in basic o qualsiasi altra cosa.

Se ben pochi utenti arriveranno mai a scrivere un programma lungo

30 o più kilobytes, pure comprendendo in questi uno screen, tali dimensioni sono invece normali per i programmi commerciali, ed è palese che trasferendoli su cartridge si ottengono notevoli vantaggi pratici, sia che si tratti di giochi, che di «gestionali» (word-processor, archivio o altro).

Quando si cerca di operare il tra-

sferimento, però, ci si scontra spesso con varie difficoltà, dovute vuoi a sistemi di protezione, vuoi alla lunghezza del software considerato, che in questo caso funziona ancora come una forma di protezione; nell'articolo che segue cerchiamo di analizzare questi ostacoli, suggerendo alcuni modi per aggirarli.

Premesso che l'argomento è vasto e presenta una casistica molto diversificata, ragione per cui ci riserviamo di continuare il discorso in seguito, bisogna innanzitutto verificare se il trasferimento può essere fatto: vi sono infatti alcune situazioni decisamente impossibili. Vediamo quali sono i casi che si possono incontrare:

- a) solo *program*
- b) solo *bytes*
- c) *program + bytes*
- d) *program + bytes + dati senza header*
- e) *program + dati senza header*
- f) *bytes + dati senza header*

```

10 CLEAR 32749
20 FOR I=32750 TO 32763
30 READ M: POKE I,M: NEXT I
40 DATA 55,62,0,221,33,0,125,1
7,20,0,205,84,5,201
50 CLS: PRINT AT 10,2;"Avvia
il registratore"
60 PRINT: PRINT TAB 4;"Premi
BREAK per fermare"
70 RANDOMIZE USR 32750
80 CLS: PRINT "*****
*****" D
ATI HEADER"*****
*****"
90 PRINT AT 4,0;"Identita' dat
i:"
100 PRINT AT 6,0;"Nome file:"
110 LET Q=PEEK 32000: PRINT AT
4,16;"BASIC" AND Q=0;"MATR.NUM."
AND Q=1;"MATR.ALFA." AND Q=2;"
BYTES" AND Q=3
120 PRINT AT 6,11:
130 FOR B=32001 TO 32010
140 PRINT CHR$ PEEK B;
150 NEXT B
160 LET A=PEEK 32011: LET B=PEE
K 32012
170 LET N=A+B*256
180 LET C=PEEK 32013: LET D=PEE
K 32014
190 LET M=C+D*256
200 LET E=PEEK 32015: LET F=PEE
K 32016
210 LET G=E+F*256
220 PRINT AT 8,0;"Lungh.cod.:"
;n
230 PRINT AT 10,0;"Byte partenz
a:";m
240 IF PEEK 32000=0 THEN PRINT
AT 10,0;"Autostart linea:";m;A
T 12,0;"Lungh.basic.:";o
250 PRINT AT 20,0;"Altri dati?(
s/n)"
260 IF INKEY$="" THEN GO TO 26
0
270 IF INKEY$="s" OR INKEY$="S"
THEN GO TO 50
280 IF INKEY$<>"s" OR INKEY$<>"
S" THEN STOP
290 STOP

```

In ogni caso, un header, almeno sul primo blocchetto di dati, esiste sempre, ed è buona norma analizzarlo, anche quando (caso a) le cose sembrano lineari.

L'header è un gruppo di pochi bytes (16) che viene caricato (o salvato) subito dopo il primo segnale di controllo, e che contiene una serie di informazioni atte a identificare univocamente il file che sta per essere caricato (o salvato). Contiene, tra l'altro, la lunghezza in bytes, la linea di autostart (se esiste) per i files basic, la locazione di inizio caricamento (per i CODE).

Normalmente, l'header viene caricato nell'area riservata al calcolatore, oltre al basic, in locazioni non identificabili, e non è più rintracciabile dopo il completamento del LOAD; non è però difficile, spostando un puntatore, caricarlo in un luogo a nostro piacimento, in modo da poterlo leggere con comodo: questo lavoro viene svolto dal programmino che trovate listato, un lettore di header, appunto (chi ci segue dai primi numeri dovrebbe trovarlo piuttosto familiare).

Per concludere i preliminari, non dovete mai dimenticare che l'inseri-

mento dell'interfaccia 1 e l'uso dei canali comportano importanti modificazioni nella mappa di memoria (cfr. i recenti articoli sull'accesso casuale ad drive e quello sui canali, in questo numero), così sintetizzabili: aggiunta di un blocco di variabili di sistema, per complessivi 58 bytes (fissi), più 32 bytes per il bit-map dei settori della cartuccia, più 595 bytes occupati temporaneamente, nella fase di trasferimento dati. Tutto ciò "a monte" del basic, che viene così spostato verso locazioni più alte.

La conseguenza fondamentale è che un CLEAR x dovrà lasciare, sotto di sé, spazio sufficiente per queste aree di lavoro. Vediamo ora singolarmente i diversi casi.

Solo program

Normalmente non ci sono problemi per trasferire un programma basic; l'unica difficoltà può essere quella di trovare la linea di autostart, rintracciabile con il lettore di header. Caricando dal registratore, si può evitare l'esecuzione dell'autostart con il MERGE: fatto questo, non dovrete più incontrare problemi, basterà salvare con

```
SAVE "m";1;"nome" LINE x.
```

Nel caso di un programma che prevede salvataggi su nastro (frequente con gli adventure), dovrete prima listarli e aggiornare la sintassi di tutti i comandi LOAD, SAVE, MERGE, VERIFY.

Solo bytes

Si distinguono due sottocasi:

1. Il blocco di dati, ci rivela l'header, inizia dalla locazione 16384, ed è lungo 49152 bytes, cioè inizia con il display-file e occupa l'intera memoria. Niente da fare, dovrete tenerlo sulla cassetta.

2. La lunghezza è tale che il caricamento può avvenire dall'indirizzo 25000 (circa), senza uscire da RAM-TOP. L'operazione in questo caso è possibile.

Program + bytes

La prima parte, il program, serve come caricatore primario della parte in linguaggio macchina, nonché da protezione e start. In molti casi è sufficiente il solito MERGE per le-



varci dai pasticci, ma bisogna fare attenzione alla presenza di eventuali CLEAR, che potrebbero non lasciare spazio per il trasferimento dei dati.

Nel casi in cui il program carica anche uno screen (la maggioranza dei giochi), se volete conservarlo si può sempre caricare a parte e trasportare sulla cartuccia, anche se ci sembra uno spreco di nastro.

Prima di essere trasferita sul microdrive, la parte in basic (punteggiatura anche le ovvietà) va opportunamente modificata nelle istruzioni di LOAD, altrimenti cercherà ancora di caricare dal registratore.

La decodifica degli header ci darà le informazioni necessarie sulla posizione dell'autostart, l'inizio del blocco CODE, la sua lunghezza.

Program + bytes + sez. senza header

In questo caso il program serve come caricatore primario di una sezione di bytes, ed eventualmente di uno screen. La sezione bytes, a sua volta, funge da caricatore secondario e da start del programma vero e proprio, in ling. macchina. Il discorso qui si complica un poco.

In molti casi, le parti program e bytes, dopo l'analisi, dovranno essere sostituite: occorrerà scrivere un nuovo breve program e uno o più falsi header, per poter caricare le parti che ne sono prive.

Con il lettore di header scopriamo a che locazione viene memorizzato il caricatore secondario; quindi dobbiamo caricare quest'ultimo e passarlo con un disassemblatore; le istruzioni da individuare sono le seguenti:

```
LD DE, lung      17 nn nn
LD IX, init      221 33 nn nn
LD A, 255        62 255
SCF              55
CALL 1366        205 86 5
RET              201
```

In cui lung è il numero di bytes da caricare E init l'indirizzo da cui iniziare il LOAD. La sequenza può avere delle varianti: anziché LD A,255 potete trovare XOR A (codice 175); le prime tre istruzioni possono essere in un altro ordine; in luogo delle ultime due può esserci JP 1366 (195 86 5).

Se si trova l'istruzione XOR A, significa che i dati sono caricati sotto forma di header, nell'altro caso l'insieme è visto come un blocco di dati normale; inutile dire che il primo crea maggiori problemi.

Dalle coppie di registri DE e IX otteniamo le informazioni contenute solitamente nell'header; dopo aver verificato che il programma non sia tanto lungo da non lasciare spazio all'area di lavoro del microdrive, dobbiamo costruire un falso header. Con una cassetta libera nel registratore, digitiamo:

```
SAVE "nome"CODE (IX), (DE)
e registriamo solo l'header, fermando subito il nastro; dopo un reset dello Spectrum, eseguiamo:
LOAD ""CODE: SAVE "m"; 1;
"nome" CODE (IX), (DE)
```

facciamo partire il nastro con il falso header che abbiamo registrato, quindi, non appena il computer l'ha "preso", sostituiamo la cassetta con quella contenente i dati da trasferire, che ovviamente doveva essere stata in precedenza posizionata correttamente.

Sul microdrive dovremo poi registrare una sezione di caricamento

in basic, con tanti LOAD quante sono le parti salvate del programma originale, ciascuno con l'adeguata sintassi.

Il caricatore secondario dovrebbe avere, dopo la sezione di caricamento che abbiamo descritto, un comando di start, in genere un salto (JP), che nel nuovo caricatore in basic diventerà una chiamata USR.

Una ulteriore complicazione può venire dal fatto che il caricatore secondario contiene qualcosa di non eliminabile, una piccola parte del programma più o meno camuffata: è necessario allora anche questa sezione, avendo cura di disattivare la parte che si occupa del caricamento, mediante la sostituzione dei codici con NOP (zeri).

Program + dati senza header

Un caso come questo in genere ha il caricatore secondario abilmente nascosto nelle variabili numeriche, o in alcuni DATA (non pretendiamo di conoscere tutte le varianti, anche perché ne vengono continuamente ideate di nuove).

Con un disassemblatore o un monitor si dovrebbe riuscire a scandagliarlo fino a rintracciare le informazioni utili, organizzate come nel caso precedente. Il modo di procedere è quindi lo stesso.

Bytes + dati senza header

Si opera ancora come nel caso precedente, l'importante è decodificare correttamente il caricatore che, attenzione, anche se registrato come bytes può contenere ugualmente un gruppo di istruzioni basic.

STRUTTURA DI UN HEADER

num.bytes	01	02-11	12-13	14-15	16-17
contenuto	tipo di file (1)	nome del file	lunghezza in bytes (2)	start (linea o address)	lunghezza in bytes (3)

- (1) 00=basic, 01=array numerico, 02=array alfab., 03=bytes
 (2) per il basic, programma + variabili
 (3) solo per il basic.





Ian Stewart & Robin Jones
Il linguaggio macchina dello Spectrum
Franco Muzzio & C. Editore, 1984 - p. 152 - L. 16.000

Un'Introduzione al linguaggio macchina del microprocessore Z80, pensato approssimato per lo Spectrum e destinato ai principianti.

Anche se la programmazione dello Z80 resta sostanzialmente la stessa per qualsiasi computer che lo utilizzi, se le procedure da eseguire, soprattutto agli inizi, tengono conto della specifica macchina le cose risultano senza dubbio più facili.

Si inizia con il solito capitolo sui sistemi di numerazione; segue un primo approccio ai registri e all'architettura del microprocessore, considerandone dapprima una versione semplificata, per illustrare modi di indirizzamento, indicizzazione, contatore di programma, stack pointer.

Quindi vengono analizzate le istruzioni di caricamento dei registri (LD), il modo di scrivere, memorizzare, salvare e ricaricare un breve programma in linguaggio macchina, con un programma basic per caricare i codici esadecimali.

Una serie di esempi di brevi routines servono poi ad affrontare una dopo l'altra le principali istruzioni, con particolare attenzione alla gestione grafica dello schermo (inevitabile, dato che il sogno della maggioranza dei programmatori in erba non è di scrivere un database ma un videogame), ai flag, ai trasferimenti di blocchi. In appendice un'utile serie di tabelle.

G. Kane
Il manuale MC68000
McGraw-Hill, 1985 - p. 168 - L. 16.000

Uscita tempestiva e quanto mai opportuna per questo volume che introduce alle meraviglie della famiglia di microprocessori attualmente sulla cresta dell'onda.

Il Motorola 68008 che, ormai lo sapete tutti, è il processore del QL, è solo uno dei membri del gruppo, che comprende 68000, 68010, 68020.

Utilizzano questa CPU, tra gli altri, l'Apple McIntosh e lo Hewlett-Packard 9816. L'ultimo nato, il 68020, è uno dei primi "veri" 32 bit, per il fatto di avere il bus dati a 32 bit reali, non multiplexati: su un solo chip, la potenza di un mainframe di pochi anni fa; ha decisamente "una marcia in più", e il libro gli dedica un capitolo a parte.

Gli altri componenti della famiglia possiedono invece lo stesso set di istruzioni: ciò significa che i programmi in linguaggio macchina scritti per il McIntosh potrebbero (in teoria) girare sul QL, e viceversa (in realtà le cose non sono così semplici).

Un manuale indispensabile, in ogni caso, per chi desidera una solida base teorica prima di lanciarsi ad assemblare con il QL; i capitoli, abbondantemente illustrati, affrontano la descrizione funzionale, la temporizzazione, le operazioni di bus, la logica di processo "exception" (cioè la gestione degli interrupt, particolarmente complessa, dato che il 68000 prevede una gerarchia di 7 livelli per le richieste esterne), i modi di indirizzamento, un cenno sugli interfacciamenti.

Preziose, infine, le appendici, con il set completo di istruzioni (24 pagine!) e i codici oggetto.

Julien Levy
Esercizi per lo Spectrum
EPSI/Editsi, 1984 - p. 116 - s.i.p.

Per chi sta muovendo i primi passi con il basic, un agile volumetto di esercizi: la prima parte enuncia i problemi, indica quale deve essere l'input e quale l'output, e analizza il tutto in un diagramma di flusso; la seconda parte propone una soluzione a ciascun esercizio, con l'aggiunta di commenti esplicativi a tutti i passaggi che possono risultare di difficile comprensione.

Gli esercizi sono suddivisi in diversi gruppi: stampa e semplici calcoli, trattamento di matrici, trattamento di stringhe, istogrammi, grafica, lavagna elettronica, grafici di funzioni, animazioni, giochi.

Per il taglio evidentemente didattico, il libro potrebbe interessare, più che ai singoli utenti, ai conduttori di corsi di informatica di primo livello.



VIDEOREGISTRI?

VR insegna, aggiorna
ti fa toccare con mano
tutte le novità

VR
VIDEOREGISTRARE

IL MENSILE DI VIDEOREGISTRAZIONE CREATIVA, TV
& COMPUTER PER TUTTI

Sped. abb. postale - Gruppo 0870 - Anno 1 Numero 1 - Maggio 85 - L. 4.000

**SPECIALE
PORTATILI:**
come si scelgono
come si usano

IN VIAGGIO CON IL VCR
le mete
da non perdere

COMPUTER:
il vostro monoscopio
personale
con il Commodore 64



OGNI MESE IN EDICOLA



DIGITARE STANCA



DIGITARE STANCA!

I programmi più interessanti spesso sono molto lunghi, un listato pubblicato è faticoso da leggere...

Sinclair Computer vi offre un'alternativa: le cassette con tutti i programmi pubblicati sulla rivista.

Ogni nastro contiene il software di un numero di Sinclair Computer, a un prezzo incredibilmente basso: solo 5.800 lire (+ 1.000 lire per spese di spedizione).

Riceverete le cassette direttamente a casa vostra, utilizzando il coupon qui a fianco.

DIGITARE STANCA è un'iniziativa

S systems

Desidero ricevere le cassette con il software pubblicato sui seguenti numeri di Sinclair Computer:

importo L.
spese di spedizione L. 1.000

Totale L.

ho versato l'importo sul c/c postale n. 37952207 (allego fotocopia della ricevuta di versamento)
 acclude assegno non trasf.

n. (banca)
intestato a SYSTEMS Editoriale
V.le Famagosta 75, 20142 MILANO

nome

cognome

via

CAP/città

Ritagliare e spedire in busta: a Systems Editoriale v.le Famagosta 75, 20142 Milano.

ABBONATEVI A SINCLAIR COMPUTER

Molti sinclaristi saranno rimasti sorpresi, dopo aver collegato l'Interface 1 allo Spectrum, nello scoprire che il loro computer usa per trasmettere e ricevere dati un sistema a canali; e saranno rimasti ancor più sorpresi nel vedere che questo sistema veniva usato anche senza l'espansione, sebbene il manuale non ne parli minimamente. Gli unici indizi dell'esistenza dei canali sono un'area della mappa di memoria, denominata informazioni di canale e puntata dalla variabile CHANS, e una serie di variabili, raggruppate sotto la dicitura STRMS, che occupano 38 bytes, a partire da 23568.

Canali fisici e canali logici

Un computer ha bisogno di ricevere dall'esterno dati da elaborare e di mandare all'esterno dati già elaborati. Per fare questo usa dispositivi detti periferiche che, a seconda del tipo, sono in grado di trasmettere dati al computer, o di riceverne, o entrambe le cose: per questo motivo sono anche dette dispositivi di INGRESSO/USCITA, o INPUT/OUTPUT o, brevemente I/O.

Per colloqu岸are con le periferiche, il sistema operativo deve essere in grado di indirizzarle correttamente, di pilotarle con opportuni programmi, e di riservare eventualmente aree di memoria, per usarle come buffer durante l'input/output dei dati. Tutto questo insieme di cose (periferica, routines di input/output, buffer) forma il cosiddetto canale fisico. Ogni canale fisico è costruito su misura per il tipo di periferica che deve pilotare: infatti ogni periferica necessita di particolari informazioni e routines per funzionare correttamente.

Per facilitare l'individuazione dei vari canali fisici, nello Spectrum è previsto l'uso per ogni canale di un'etichetta mnemonica, l'identificatore di canale, rappresentata da una lettera d'alfabeto.

Senza interface 1 sono ammessi solo 3 identificatori di canale:

K per la tastiera (Keyboard)

S per lo schermo (Screen)

P per la stampante ZX o simili (Printer)

I canali dello Spectrum

di Renzo Zonin

Esiste un quarto identificatore, "R", usato dal sistema operativo per scopi interni (tipicamente per scrivere nella work-space area), ma non è accessibile da basic.

Quando è collegata l'interface 1, diventano operativi altri 4 identificatori di canale:

M per i microdrives (Microdrive)

N per la rete locale (Network)

B per la RS232 modo binario (Binary)

T per la RS232 modo testo (Text)

I dati relativi ai canali fisici sono memorizzati nella zona di memoria puntata dalla variabile CHANS. Per ogni canale vengono memorizzati gli indirizzi delle due routines di gestione dell'I/O, l'identificatore di canale fisico e una serie di informazioni che dipendono dal tipo di canale. Queste informazioni possono occupare più o meno spazio; per questo l'area delle informazioni di canale è totalmente dinamica.

Per poter puntare alle informazioni relative a un canale fisico, il sistema usa le variabili di sistema chiamate STRMS. Si tratta di 38 bytes, che a 2 a 2 formano 19 indirizzi, ognuno dei quali punta alle informazioni di un canale, nell'area puntata da CHANS. In pratica, a ogni coppia di locazioni è associato un cosiddetto canale logico, che è quindi un'entità astratta, contraddistinta da un numero fra -3 e 15.

L'utente può assegnare un qualsiasi canale logico a un qualsiasi canale fisico, e da quel momento sarà il sistema operativo a occuparsi del canale fisico, mentre il programmatore vedrà solo il canale lo-

gico. Questo metodo permette di svincolare il programmatore dell'uso diretto dei canali fisici.

Configurazione standard

All'accensione il sistema operativo collega automaticamente alcuni canali logici con alcuni canali fisici, perchè in caso contrario non potrebbe colloqu岸are con l'esterno. I canali collegati per default sono:

C. Logico	C. Fisico
-3	K
-2	S
-1	R
0	K
1	K
2	S
3	P

I canali logici indicati da numeri negativi (-3, -2, -1) sono riservati al sistema operativo, e non sono visibili né accessibili dal basic. Ecco perchè lo Spectrum riesce a mostrarci sempre i suoi messaggi di errore, anche nel caso noi avessimo ridefinito i canali (almeno finchè non lo mandiamo in crash...)

I canali dallo 0 in poi sono invece disponibili al programmatore da basic. Lo 0 riceve dati dalla tastiera, e scrive nella parte bassa dello schermo.

Il canale 1 si comporta nello stesso modo (vedremo che è collegato allo stesso canale fisico).

Il canale 2 scrive nella parte alta dello schermo e non ha funzioni di input.

Il canale 3 scrive sulla stampante

ZX; anch'esso non ha funzioni di input.

Le normali istruzioni del basic vanno a scrivere sui canali di default, e nascondono in questo modo agli occhi del programmatore tutto il lavoro svolto dal sistema. Quando in un programma scriviamo PRINT, per il sistema è come se avessimo scritto PRINT # 2: il cancelletto è il prefisso per "canale", mentre 2 è il numero di canale logico. Allo stesso modo, LPRINT equivale a PRINT # 3 e INPUT a INPUT # 0.

Che cosa succede se cambiamo i canali di default?

Modifiche alla configurazione standard

La prima cosa che possiamo provare (anche sullo Spectrum inespanso) è la modifica *temporanea* del canale di default. Per esempio, provate a dare

```
LPRINT# 2; "prova"
```

La scritta apparirà nella parte alta dello schermo, nonostante che LPRINT sia normalmente un co-

mando per la stampante. Provate invece:

```
LPRINT# 0; "prova": PAUSE 0.
```

La scritta apparirà questa volta nella parte bassa dello schermo. Il PAUSE 0 serve a impedire che la scritta venga cancellata dal messaggio di OK.

Capito il meccanismo, possiamo sfruttarlo per qualcosa di pratico. Supponiamo di voler scrivere un programma che si possa usare a scelta con uscita su schermo o su stampante ZX: definiremo una variabile *s* uguale a 2 per uscita su schermo e uguale a 3 per uscita su stampante. Per esempio:

```
INPUT "Screen or Printer (S/P)";  
R$  
IF R$ = "S" THEN LET S = 2  
IF R$ = "P" THEN LET S = 3  
PRINT# S; "PROVA"  
RETURN
```

E se invece avessimo un programma già pronto che lavora su video, e volessimo la stampa dell'esecuzione? O si modificano tutte le istruzioni PRINT del programma op-

pure, più semplicemente, si cambia stabilmente il settaggio dei canali; la modifica è permanente, fino a un nuovo comando, o spegnimento, o al comando CLEAR #, che rimette le cose in condizioni standard.

Per modificare permanentemente l'assegnazione di un canale logico lo si deve riaprire, specificando il nuovo canale fisico da collegare. Nel caso che stiamo considerando, si dovrà dare

```
OPEN# 2, "p"
```

Per riportare alle condizioni iniziali uno specifico canale fra lo 0 e il 3 potete usare il comando CLOSE #*n*, con *n* numero del canale: questo lo riporta alle condizioni iniziali.

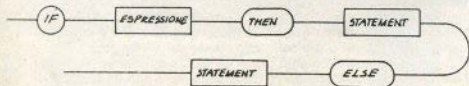
Conclusione

Avete visto come si possono sfruttare in modo atipico i canali dello Spectrum standard. Nella prossima puntata illustreremo più in dettaglio le aree di memoria destinate ai canali, e vedremo l'uso dei canali aggiunti dall'interfaccia 1.

Un po' di Pascal

a cura di Monica Fumagalli

FIG. 11 - IF STATEMENT



Strutture condizionali

IF STATEMENT (Fig. 11)

Un *if statement* permette l'esecuzione di uno statement, o di una sequenza di statements, condizionata alla valutazione di una espressione booleana.

La parte ELSE può essere mancante e in questo caso, se il risultato della valutazione dell'espressione è *false*, non viene compiuta nessuna azione. In ogni caso l'esecuzione prosegue con lo statement che segue l'*if statement*.

Gli statements che seguono le parole chiave THEN e ELSE possono essere a loro volta *if statements*: in questo caso bisogna fare attenzione al matching dell'*else*; il PASCAL utilizza il matching implicito dell'*else* con l'*if* più interno; se il matching che si desidera ottenere è differente bisogna utilizzare esplicitamente il compound statement.



CASE STATEMENT (Fig. 12)

Un *case statement* permette di selezionare differenti (compound) statements, dipendentemente dal risultato della valutazione di una espressione scalare. Ovviamente tutti i valori della lista dei label devono essere dello stesso tipo del risultato della valutazione dell'espressione; ogni valore può appartenere a un solo label. Per chiarire la struttura case vediamo alcuni esempi:

```
VAR mese: 1..12;  
    giorni: 28..31;  
    ...
```

CASE mese OF

```
11, 4, 6, 9, : giorni := 30;  
2 : IF annobis (x) THEN  
    giorni := 28
```

ELSE

```
    giorni := 29;  
13, 5, 7, 8, 10, 12 : giorni := 31  
END
```

CASE comando OF

```
'I' : inserimento;  
'C' : cancellazione;  
'S' : sostituzione;  
'F' : fine  
END
```

Quando un case statement viene eseguito, si valuta l'espressione e si esegue solo lo statement il cui label contiene il risultato della valutazione; se nessun label contiene questo valore, viene dato messaggio di errore.

Strutture iterative

WHILE STATEMENT (Fig. 13)

Un *while statement* indica che uno o più statements devono essere eseguiti ripetutamente, fino a quando una certa condizione rimane verificata. Questa espressione viene valutata prima di ogni iterazione, per decidere se proseguire o se trasferire il controllo allo statement che segue il while. In particolare, se l'espressione risulta falsa alla prima valutazione, il while non viene mai eseguito. Vediamo alcuni esempi:

```
VAR F : FILE OF info;  
    ...  
RESET (F);  
WHILE NOT EOF (F) DO
```

FIG. 12- CASE STATEMENT

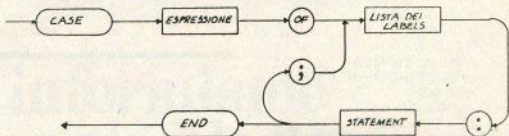


FIG. 13- WHILE STATEMENT

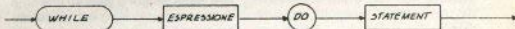


FIG. 14- REPEAT STATEMENT

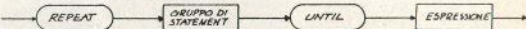
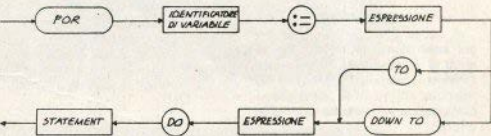


FIG. 15- FOR STATEMENT



```

BEGIN
  eseguiopsu (F );
  get (F)
END

VAR tabella: ARRAY [1..max] OF
Info;
  chiave : info;
  trovato : boolean;
  i : integer;
...
i := 0;
trovato := false;
WHILE (NOT trovato) AND (i =
max) DO
  BEGIN
    i := i + 1;
    trovato := tabella [i] = chia-
ve
  END

```

REPEAT STATEMENT (Fig. 14)

Un *repeat statement* permette di indicare che uno o più statements devono essere eseguiti ripetutamente, fino a quando una data condizione diventa verificata. Questa espressione viene valutata al termine di ogni iterazione, quindi anche se è verificata il repeat viene eseguito, a differenza del while, almeno una volta. Vediamo un esempio:

```

VAR F : FILE OF Info;
filevuoto : boolean;
...
IF EOF (F) THEN
filevuoto := true
ELSE
  REPEAT
    eseguiopsu (F );
    get (F)
  UNTIL EOF (F)

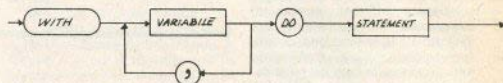
```

A causa di quanto detto sopra sull'esecuzione del *WHILE statement*, è necessario trattare separatamente il caso di un file inizialmente vuoto, altrimenti la prima esecuzione del repeat darebbe luogo a un errore.

FOR STATEMENT (Fig. 15)

Un *for statement* permette di indicare che uno o più statements devono essere eseguiti un fissato numero di volte. La variabile di controllo che determina il numero delle iterazioni deve essere di tipo scalare. Le parole chiave TO e DOWNTO sono usate alternativamente, nel caso in cui rispettivamente il valore iniziale sia minore o maggiore di quello fi-

FIG. 16 - WITH STATEMENT



nale.

La variabile di controllo viene incrementata (o decrementata) automaticamente, e dopo il completamento del *for* il suo valore è indefinito. I valori delle espressioni iniziali e finale vengono valutati una sola volta, e non possono quindi essere modificati durante l'esecuzione del *for*.

WITH STATEMENT (Fig. 16)

Questo statement non fa parte né degli statement eseguibili né delle strutture di controllo: ha come unico scopo quello di facilitare l'utilizzo delle variabili di tipo *record*.

Infatti un *with statement* permette di accedere direttamente ai campi di un record senza dover utilizzare la normale sintassi cioè *identificatore-record.identificatore-campo*. Vediamo un esempio:

```

VAR persona : RECORD
  età, nrscarpe: integer;
  sesso: char;
  colcap: (rossi, blondi, neri,
castani);
  colocchi: (azzurri, marroni)
END;

```

```

...
WITH persona DO
  BEGIN

```

```

    età = 23;
    nr scarpe = 41;
    sesso = 'm';
    colcap = 'blondi';
    colocchi = 'azzurri'
  END

```

Se il *with* è seguito da più variabili, si può creare una ambiguità sintattica se campi di record diversi hanno lo stesso identificatore; que-

sta ambiguità viene risolta dando la priorità al campo del record del *with* più interno. Infatti la scrittura:

```

WITH A,B,...,Z DO
  statement

```

è equivalente alla seguente struttura di *with* innestati:

```

WITH A DO
  WITH B DO
    ...
    WITH Z DO
      statement
    END
  END
END

```

Per chiarire questa situazione vediamo un esempio:

```

VAR REC1 : RECORD
  a, b : real
  END;
REC2 : RECORD
  b,c : real
  END;

```

```

WITH REC1, REC2 DPO
  BEGIN
    b := 78.965; (si riferisce
a REC2)
    REC1.b := 0.001;
    a := 7.77;
    c := 123.456
  END

```

Avrete notato che negli esempi non tutti gli *statement* sono seguiti dal punto e virgola: la regola generale per il separatore è la seguente: si deve digitare ";" dopo ogni *statement*, eccetto quelli che precedono le parole chiave:

```

ELSE
END
UNTIL.

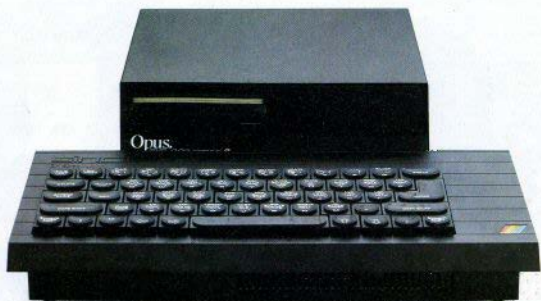
```





QUI SMAU
SMAU 1985, 19-24 SETTEMBRE
PADIGLIONE 15.2 - STAND L1/H6

**Vi informiamo
che il vostro piccolo Spectrum
è molto più grande.**



Opus.

DISCOVERY 1

**BIT SHOP
primavera**



Discovery, è l'unità a dischi più avanzata e più completa presentata fino ad ora per potenziare lo Spectrum; sfrutta i più moderni ritrovati della tecnologia nel campo dei dischetti.

Il modello più economico, Discovery 1, comprende: • Unità a dischetti standard da 3,5 pollici. • Connettore passante per periferiche. • Interfaccia joystick. • Porta per stampante parallela. • Uscita per monitor monocromatico.

Discovery 1 può essere potenziato successivamente aggiungendo un secondo disc drive per mezzo del kit di espansione "Discovery +"; in alternativa, è disponibile il "Discovery 2" già dotato di due drive.



Questo è da solo trasforma in un si



CARATTERISTICHE TECNICHE

Unità a dischetti:

Tipo del disco	standard da 3,5in
Modello 35401
Numero delle tracce 40
Numero delle facce 1
Capacità totale 250 Kbyte
Capacità formattata 180 Kbyte
Accesso traccia-traccia 3 ms
Alimentazione interna



Discovery: il vostro Spectrum stema.

Discovery è stato concepito in modo da funzionare con gli stessi comandi della "Interface 1" per permettervi l'uso, senza bisogno di modificare nulla, dei numerosi programmi disponibili su cassette o su cartucce per microdrive.

Discovery non occupa "RAM utente" dello Spectrum, per cui potete eventualmente utilizzare programmi su cassetta senza scollarlo dal computer. Superfluo diventa anche l'alimentatore Sinclair in quanto è la stessa unità (dotata di interruttore generale) ad alimentare lo Spectrum.



BIT SHOP
primavera

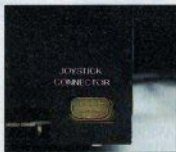


Interfacce:

- Connettore passante per allacciamento delle periferiche, da 56 poli a circuito stampato.
- Porta per stampante parallela, compatibile Centronics.
- Interfaccia joystick: connettore a 9 poli tipo "D" (compatibile Kempston)
- Uscita monitor: plug audio

Alimentazione:

- Tensione di rete: 220 V, 50 Hz



Distribuito e garantito in Italia da Bit Shop Primavera.



ABRUZZI

● **Lanciano** - DIGIT TEAM, Via Mario Bianco 2, Tel. (0872)37266 ● **Pescara** - COMPUTER MARKET, Via Trieste 73, Tel. (085)26007

CALABRIA

● **Catanzaro** - VISICOM, Via XX Settembre 62/A, Tel. (0961)24181 ● **Cosenza** - DEFIM, Via Massaua 25, Tel. (0984)74214 ● **Reggio Calabria** - PROTEO, Via S. Marco 8/B, Tel. (0965)21685

CAMPANIA

● **Benevento** - DE.VI COMPUTER, Via E. Goduti 62/64, Tel. (0824)54005 ● **Casupella** - ANDROMEDA, Via Appia 128, Tel. (0823)460469 ● **Napoli** - COMPUTER FACTORY, Via L. Giordano 40/42, Tel. (081)241242 - COMPUTER FACTORY, Via G. B. Marino 11/13, Tel. (081)621379 ● **Quarto** - S.G. CRISTOFARO, Via Campana 255bis, Tel. (081)876676 ● **Salerno** - GENERAL COMPUTER, C.so Garibaldi 56, Tel. (089)237835

EMILIA-ROMAGNA

● **Bologna** - EDP SISTEMI BOLOGNA, V.le Pietramellara 61/F, Tel. (051)263032 ● **Forlì** - Home & PERSONAL COMPUTER, P.za Melozzo 1, Tel. (0543)35209 ● **Lugo** - SELCO ELETTRONICA, Via Magnanoni 26, Tel. (0545)22661 ● **Parma** - BIT SHOW, Borgo Parente 14/E, Tel. (0521)25014 ● **Reggio Emilia** - MICROINFORMATICA, Via S. Giuseppe 4/A, Tel. (0522)347116 ● **Rimini** - COMPUTER SHOP, Via Garibaldi 44, Tel. (0541)27691 ● **Sassuolo** - MICROINFORMATICA, P.za Martiri Partigiani 31, Tel. (0536)802995

FRIULI VENEZIA GIULIA

● **Trieste** - COMPUTER SHOP, Via P. Reti 6, Tel. (040)61602 ● **Udine** - P.S. ELETTRONICA, Via Tavagnacco 89/91, Tel. (0432)482086

LAZIO

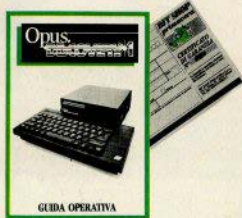
● **Rieti** - ESSEEMMECI, Via Cintia 70, Tel. (0746)44704 ● **Roma** - COMPUTER MARKET, P.za S. Donati Prave 14, Tel. (06)7945493 - EASY-BYTE, Via G. Villani 24/26, Tel. (06)7811519

LIGURIA

● **Genova** - COMPUTER CENTER, Via S. Vincenzo 109R, Tel. (010)581474 ● **Genova Sampierdarena** - COMPUTER CENTER, Via G.D. Storace 4R, Tel. (010)454107 ● **Imperia** - COMPUTER SHOP, Via A. Doria 45, Tel. (0183)275448 ● **La Spezia** - I.L. ELETTRONICA, Via Luntiana 481, Tel. (0187)511739 ● **Recco** - DIGIT CENTER, Via B. Assoreto 78, Tel. (0185)74252

LOMBARDIA

● **Bergamo** - SANDIT, Via S. Francesco D'Assisi 5, Tel. (035)224130 ● **Brescia** - IL COMPUTER, Via Solferino 5, Tel. (030)42100 ● **Busto Arsizio** - BUSTO BIT, Via Gariviana 17, Tel. (0331)825034 ● **Cesano Maderno** - ELECTRONIC CENTER, Via Ferrari 6, Tel. (0362)520728 ● **Cinisello Balsamo** - GBC ITALIANA, V.le Matteotti 66, Tel. (02)6181801 ● **Crema** - EL.COM, Via IV Novembre 56/58, Tel. (0373)83393 ● **Gallarate** - S.E.D., Via Arnaldo da Brescia 2, Tel. (0331)795735 ● **Lecco** - S.G.A. INF. FORMATICA, Via Leonardo da Vinci 7, Tel. (0341)361264 ● **Leino** - MICRO COMPUTER, V.le Amendola 48, Tel. (0332)537536 ● **Mantova** - ANTEC COMPUTER, Via Cavour 69/71, Tel. (0376)329333 ● **Milano** - GBC ITALIANA, Via Petrella 6, Tel. (02)2041051 - GBC ITALIANA, Via Cantoni 7, Tel. (02)437478 - LAS VEGAS, Galleria Manzoni 40, Tel. (02)705055 - NUOVA NEWEL,



Via Mac Mahon 75, Tel. (02)323492 ● **Monza** - EMI, Via Azzone Visconti 39, Tel. (039)388275 ● **Parva** - M3 COMPUTERS, V.le C. Battisti 4/A, Tel. (0382)31087 ● **Sesto Calende** - I.A.C. NUOVE TECNOLOGIE, Via Matteotti 38, Tel. (0331)923134 ● **Varese** - SUPERGAMES COMPUTER, Via Carrobbio 1, Tel. (0332)241092 ● **Vigevano** - VISENTIN, C.so Vittorio Emanuele 82, Tel. (0381)83833 ● **Voghera** - BYTE ELETTRONICA, Via Matteotti 65, Tel. (0382)212280

MARCHE

● **Ancona** - CESARI, Via De Gasperi 40, Tel. (071)85620 ● **Macerata** - CERQUETELLA, Via Spalato 126, Tel. (0733)35344 ● **Senigallia** - C.I.D.I., Via Maierini 10, Tel. (071)659131

MOLISE

● **Companone** - SISTEMA, Via Monsignor Bologna 10, Tel. (0874)94795 ● **Termoli** - ROSATI COMPUTER, Via Martiri della Resistenza 8, Tel. (0875)82291

PIEMONTE

● **Alba** - PERSONAL & HOME COMPUTER, Via Paruzza 2, Tel. (0173)54441 ● **Alessandria** - BIT SYSTEM, Via Savonarola 13, Tel. (0131)445692 ● **Biel-la** - NEGRINI, Via Tripoli 32/A, Tel. (015)402861 ● **Cuneo** - THEM, Via Statuto 10, Tel. (0171)60983

BIT SHOP primavera



LA PRIMA RETE DI SPECIALISTI IN PERSONAL COMPUTER.

BIT SHOP PRIMAVERA S.p.A.
VIA C. FARINI, 82, 20159 MILANO
TEL. (02)6880429/6882171

Favria Canavese - MISTER PERSONAL, Via Capor Caltanico 52, Tel. (0124)42344 ● **Torino** - A. B. COMPUTER, C.so Grossotto 209, Tel. (011)2163665 ● **COMINFOR SISTEMI**, C.so B. Telesio 4/B, Tel. (011)793007 - COMPUTER SHOP, Via Nizza 91, Tel. (011)6509576 - DUEGI, C.so Racconigi 26, Tel. (011)3358756 - ZUCCA COMPUTER, Via Tripoli 179, Tel. (011)352262 ● **Verbania-Intra** - ELLIOTT COMPUTER SHOP, P.zza Don Minzoni 32, Tel. (0323)435517

PUGLIA

● **Bari** - NUOVA HALET ELECTRONICS, Via Estramurale Capruzzi 192, Tel. (080)228822 - TECNOSISTEMI, P.zza Garibaldi 55, Tel. (080)210084 ● **Foggia** - SINFOR, V.le Europa 44/46, Tel. (0881)32579 ● **Lecce** - CAMEL ELETTRONICA E COMPUTER, Via Marinucci 1/3, Tel. (0832)592861 ● **Taranto** - SUD-COMPUTER, Via Polibio 7/A, Tel. (099)338041

SARDEGNA

● **Cagliari** - BIT SHOP, Via Zagabria 47, Tel. (070)490954

SICILIA

● **Caltanissetta** - ELEONORI & AMICO, C.so Umberto 1254/256, Tel. (0934)25610 ● **Catania** - ARIA NUOVA, P.zza Lanza 16, Tel. (095)438573 ● **Messina** - TEMPO REALE, Via del Vespro 71, Tel. (090)773983 ● **Palermo** - INFORMATICA COMMERCIALE, Via Notarbartolo 23/B, Tel. (091)291500 ● **Siracusa** - LOGOL SYSTEM, V.le Scala Greca 339/9, Tel. (0931)532244

TOSCANA

● **Firenze** - ANDREI CARLO & C., Via G. Milanese 28/30, Tel. (055)472810 - ELETTRONICA, CENTOSTELLE, Via Costantello 5/A, Tel. (055)610251 ● **Livorno** - C.P.A. ELETTRONICA, Via Paoli 32, Tel. (0586)27357 ● **Lucca** - LOGOS INFORMATICA, V.le S. Concordio 160, Tel. (0583)55519 ● **Pisa** - IT-LAB, Via XXIV Maggio 101, Tel. (050)552590 - TECNINOVA COMPUTER, Via Emilia 36, Tel. (050)502516 ● **Pistoia** - C.D.E. V.le Adun 350, Tel. (0573)400712 ● **Prato** - C. BARAGLI ELETTRONICA, Via E. Boni 76/80, Tel. (0574)595001 ● **Viareggio** - C.D.E., Via A. Volta 79, Tel. (0584)942244

TRENTINO ALTO ADIGE

● **Bolzano** - BONTADI, P.zza Verdi 15/B, Tel. (0471)971619 - ROVERETO - S.E.D.A., Via Fontana 8/B, Tel. (0464)34506 ● **Trento** - S.E.D.A., Via Sighele 7/1, Tel. (0461)996546

UMBRIA

● **Perugia** - STUDIO SYSTEM, Via D'Andreotto 49/55, Tel. (075)754964

VALLE D'AOSTA

● **Aosta** - INFORMATIQUE, Av. Du Conseil des Commis 16, Tel. (0165)362242

VENETO

● **Bussano del Grappa** - TODARO, Via Jacopo da Ponte 51, Tel. (0424)22810 ● **Mirano** - SAVING ELETTRONICA, Via Gramsci 40, Tel. (041)432876 ● **Padova** - COMPUTER POINT, Via Roma 63, Tel. (049)22564 - S.I.C. ITALIA, Via Nicolo Tommaseo 13, Tel. (049)663133 ● **San Donà di Piave** - TIME, P.zza Rizzo 61, Tel. (0421)50474 ● **CHIP COMPUTER**, Stradone S. Felice, Tel. (0445)21255 ● **Vicenza** - A.T.R., Via del pro. Tel. (0444)56461



Tecniche di indirizzamento

I modi di indirizzamento del microprocessore Z80 non sempre risultano efficienti; talvolta si rende necessaria un'opportuna combinazione dei modi disponibili, al fine di simularne altri, tipicamente riscontrabili in microprocessori piú evoluti (per un elenco dei modi di indirizzamento dello Z80) si veda SC n. 10 a pag. 15.

Indirizzamento indicizzato (con indice a 16 bit)

Per simulare questo modo di indirizzamento si usa l'istruzione

```
ADD HL, rr
```

si ottiene in HL l'indirizzo indicizzato (la somma dell'indirizzo base con l'indice a 16 bit contenuto nella coppia di registri rr). Es: per simulare una istruzione del tipo

```
LD A, (HL+DE)
```

dove DE contiene l'Indice:

```
LD HL, TABLE ;indirizzo base
LD DE, 312 ;indice
ADD HL, DE ;HL=indir.indicizzato
LD A, (HL) ;preleva il dato
```

Indirizzamento indicizzato con indice e offset

In questo caso si unisce alla tecnica appena vista la possibilitá di indicizzazione dello Z80. Il seguente esempio simula un'istruzione del tipo

```
LD A, (HL+DE+n)
dove DE contiene l'Indice a 16 bit
e n definisce un offset a 8 bit.
```

```
LD IX, TABLE ;indirizzo base
LD DE, 312 ;indice
ADD IX, DE ;IX=indir.indicizzato
LD A, (IX+22) ;prel.dato (22=offset)
```

Autopostincremento

Si tratta di un modo di indirizzamento che incrementa automatica-

Programmazione in Assembly con lo Spectrum

di Gianluca Carrì

mente dopo l'uso, il registro che punta al dato. La realizzazione per lo Z80 è elementare, dato che basta aggiungere un'istruzione di incremento:

```
LD A, (HL) ;Preleva dato
INC HL ;Incrementa HL
```

Autopredecremento

È il modo opposto al precedente: il registro che punta al dato è decrementato prima dell'uso. Ancora la realizzazione è molto semplice:

```
DEC HL ;Decrementa HL
LD A, (HL) ;Preleva dato
```

Indirizzamento indiretto preindizzato

L'indirizzo del dato, che qui chiameremo DATADDR, è localizzato all'interno di una tabella che inizia a BASE. Un registro (es. l'accumulatore) fornisce l'indice necessario a indirizzare DATADDR.

Es.:

```
LD A, 2 ;A=indice
ADD A, A ;A=A*2 (ogni indirizzo occupa 2 bytes)
LD C, A ;BC=A
LD B, 0
LD HL, BASE ;base tabella
ADD HI, BC ;indirizza DATADDR
LD C, (HL) ;preleva DATADDR in BC
INC HL
LD B, (HL)
LD A, (BC) ;preleva il dato puntato da DATADDR
```

La tabella potrebbe be seguente:

```
BASE DEFM DATAB
```

```
DEFM DATA1
DEFM DATADDR ;dato prelevato (BASE+2*2)
DEFM DATA3
DEFM DATA4
```

Indirizzamento indiretto postindizzato

In questo caso l'indirizzo prelevato indirettamente sarà usato come contenuto del registro indice necessario a prelevare il dato. Es.:

```
LD IX, (BASE) ;preleva indirizzo base
LD A, (IX+2) ;preleva dato
```

Scambio di parametri

Un problema tipico del programmatore Assembly è quello del passaggio dei parametri dal programma principale alle subroutine. Se i parametri richiesti dalle subroutine sono pochi, la soluzione piú usata e piú efficiente è quella di memorizzarli nei registri del processore. Purtroppo la soluzione non è applicabile qualora i parametri siano in numero superiore a quello dei registri disponibili.

Un metodo molto efficiente consiste nel far seguire fisicamente i parametri all'istruzione di chiamata della subroutine. Es.:

```
CALL PRINT
DEFM "Buongiorno a tutti voi!"
DEFB 0
..istruzione successiva ..
```

Dato che l'istruzione CALL me-



morizza nello stack l'indirizzo che segue l'istruzione stessa (in questo caso l'indirizzo del messaggio), la subroutine chiamata potrà facilmente prelevare le singole lettere (es. visualizzandole sullo schermo):

```
STAMPA:
POP HL      ;HL punta al messaggio
LD A,(HL)  ;preleva un carattere
INC HL     ;incrementa ind. ritorno
PUSH HL
OR A
RET Z      ;ritorna al programma
           ;principale se A=0
RST 10h   ;visualizza carattere
JR STAMPA
```

Si noti la necessità di aggiornare l'indirizzo di ritorno presente sulla cima dello stack, in modo da riprendere l'esecuzione del programma a partire dalla prima istruzione che segue il blocco di dati. In questo esempio, il byte 0 è usato come segnale di fine messaggio (*end marker*).

Un altro metodo consiste nel me-

morizzare i parametri nello stack.

Es.:

```
LD HL,DAT01 ;dato 1
PUSH HL     ;nello stack
LD HL,DAT02 ;coae sopra
PUSH HL
LD HL,DAT03 ;dato 3...
PUSH HL
LD HL,DAT04 ;dato 4...
CALL CONVERT
```

...istruzione successiva...

Dopo l'esecuzione, lo stack sarà come segue:

```
INDIRIZZO DI RITORNO
DAT04
DAT03
DAT02
DAT01
```

Ricordando che il dato sulla cima dello stack sarà quello che verrà per primo tolto da POP, la subroutine preleverà i parametri come segue:

```
CONVERT
POP HL      ;indirizzo di ritorno in HL
```

```
LD (RETURN),HL ;Memorizza brevemente
POP BC          ;BC = DAT04
POP DE          ;DE = DAT03
POP HL          ;HL = DAT02
POP IX          ;IX = DAT01
... resto della subroutine...
LD HL,(RETURN) ;Preleva indirizzo di ritorno
JP (HL)         ;Ritorna al programma principale
```

Si noti la necessità di salvare l'indirizzo di ritorno al programma principale in una locazione di memoria inutilizzata, in questo caso chiamata RETURN.

Con questa puntata termina il ciclo di articoli sulla programmazione in Assembly Z80, iniziato sul numero 4 di SC; esso è stato pensato come una semplice guida introduttiva al linguaggio; è quindi consigliabile a tutti quanti lo volessero approfondire la lettura di uno dei numerosi libri disponibili sull'argomento. Continueremo comunque a parlare di Assembly nei prossimi numeri della rivista.

CENTRO 2

ASSISTENZA HARDWARE

V. FRA CRISTOFORO, 2 - 20142 MILANO - (02) 8434368



CENTRO ASSISTENZA
AUTORIZZATO

sinclair
MICROCOMPUTER



Questo programma consente di muovere un carattere sullo schermo, superando gli inconvenienti legati alla lentezza del basic e al comando INKEY\$, che non permette di gestire la pressione contemporanea di due tasti.

Il movimento si ottiene mediante l'uso dei tasti 5-6-7-8, o di una loro combinazione per gli spostamenti diagonali; il tasto 0 consente la funzione "fire". Ovviamente, un joystick offre una maggiore praticità.

Lo scorrimento di un oggetto (della dimensione di un carattere) da una posizione all'altra dello schermo avviene, con queste routines, con movimento molto fluido, senza scatti, anche a bassa velocità, essendo ottenuto tramite shift progressivi di singole "row".

L'intero programma è composto da sette routines:

KEYTEST intercetta la pressione dei tasti e chiama in esecuzione la specifica routine di movimento;

MOVTEST controlla che lo spostamento richiesto sia consentito; se la posizione di destinazione è vietata (scontro, raggiungimento dei margini della pista o dello schermo, etc), il movimento non viene eseguito e si ottiene un rientro al basic, con codice di ritorno impostato;

WAIT è un semplice loop di ritardo, senza il quale lo scorrimento sarebbe troppo veloce e praticamente incontrollabile;

MOVNORD/MOVSUD eseguono gli spostamenti verticali e diagonali, verso l'alto e verso il basso;

MOVEST/MOVOV eseguono gli spostamenti orizzontali, verso destra e verso sinistra.

La complessità di MOVNORD / MOVSUD è dovuta non tanto all'esecuzione anche degli spostamenti diagonali, "composti", quanto alla particolare struttura del display-file nella RAM, e alla conseguente difficoltà nella gestione del video: ogni riga di stampa è suddivisa in 8 linee (row) non contigue, e le 24 righe sono a loro volta mappate in tre sezioni di memoria: ciò rende piuttosto complesso il controllo degli spostamenti verticali, soprattutto nel passaggio da un'azione all'altra.

Il movimento sullo schermo

di Giovanni Mellina

Per verificare la validità di uno spostamento, la routine MOVTEST controlla che la posizione che si tenta di occupare sullo schermo non sia vietata, in base a una regola, che occorre stabilire al momento dell'utilizzo del programma; per evitare ulteriori complicazioni, tale regola dovrà essere il più possibile elementare: nel programma presentato, lo spostamento viene impedito dalla presenza di attributo 0 (INK 0, PAPER 0, BRIGHT 0, FLASH 0) nella posizione di destinazione.

Si possono definire altre condizioni, modificando gli statements 73-75; per esempio:

a) spostamento non consentito con attributo BRIGHT = 1

```
73 eliminare
74 BIT 6,A          203 119
75 RET Z           200
```

b) spost. non consentito con INK diverso da 0

```
73 eliminare
74 AND 7           230 7
75 RET Z           200
```

c) spost. non consentito con attributo diverso da 58 (= INK 2 + PAPER 7 + BRIGHT 0 + FLASH 0)

```
73 eliminare
74 XOR 58          238 58
75 RET Z           200
```

E così via: per stabilire altre condizioni, dovete solo tenere presente il valore degli attributi:

```
FLASH 1 = 128
BRIGHT 1 = 64
PAPER x = x*8
INK y = y.
```

Quando si cerca di eseguire uno spostamento vietato, si ottiene un immediato rientro al programma basic, con assegnazione al codice di

ritorno del valore dell'attributo presente nella posizione vietata: è così possibile far proseguire il programma in modo diversificato, in conseguenza di tale valore.

Analogamente, la pressione del tasto 0 funge da fire, e comporta sempre il ritorno al basic (o il salto ad altra routine), con codice di ritorno 999.

Per un corretto controllo di questi return codes, è necessario che la chiamata al linguaggio macchina avvenga con

```
LET var = USR Indirizzo
```

in cui var sia una qualsiasi variabile numerica, che al rientro verrà impostata con il contenuto del registro BC (appunto il return code), in modo da poter essere testata, per esempio con

```
IF var = valore THEN ...
```

La posizione in cui si trova il carattere comandato dai cursori viene costantemente tenuta aggiornata: nei bytes 23728/29 (area variabili di sistema, locazioni non usate dallo Spectrum) è memorizzato l'effettivo indirizzo in display-file, mentre agli indirizzi 23296/97 (primi due bytes del buffer stampante) si trovano rispettivamente i numeri di riga e di colonna (coordinate AT y,x); questi valori possono essere acquisiti dal programma basic semplicemente con PEEK.

La routine WAIT controlla la velocità di scorrimento: è predisposta con un coefficiente di rallentamento adatto a giocatori di media abilità, che può essere facilmente cambiato, alterando il valore 3 allo statement 80 (indirizzo di caricamento + 142). Diminuendo si accelera e





umentando si rallenta.

Per una regolazione piú fine della velocitá, intervenite sull'istruzione successiva (Indir. di caric. + 144), sostituendo il valore 0: potete varlarre da 1 a 255, con l'incremento rallenta e viceversa.

Evidentemente, questi valori possono essere variati automaticamente dal programma (con POKE), per esempio per aumentare progressivamente la velocitá da una *manche* all'altra.

Prima di eseguire la chiamata al programma in codice macchina, occorre impostare le coordinate di schermo e l'indirizzo in display-file in cui si trova inizialmente l'oggetto (il carattere) da muovere; questo si ottiene con:

```
100 POKE 23296,y: REM y=num.
riga, 0-23
110 POKE 23297,x: REM x=num.
colonna, 0-31
120 LET w=16384+x+32*(y-INT (y/
8)+8)+2848*INT (y/8)
130 POKE 23729,INT (w/256)
140 POKE 23728,w-256*PEEK 23729
```

Per riassumere, ecco la sequenza delle operazioni:

- disegnare l'area di gioco, con sbarramenti, piste, etc.
- eseguire le POKE 23296/97 e 23728/29, con le coordinate e l'indirizzo di start
- eseguire un'eventuale POKE all'indirizzo di caricamento +142 e/o +144, per la regolazione della velocitá
- PRINT AT PEEK 23296, PEEK 23297; "\$", in cui "\$" è il carattere da muovere
- iniziare il gioco con LET rc = USR Indirizzo di caricam.

IF rc = 999 THEN GO TO gestione "fire"

IF rc = attributo THEN GO TO gestione scontro

Il semplice programma allegato alle routines in linguaggio macchina ve ne mostrerá l'uso pratico. Il programma di caricamento colloca il codice macchina all'indirizzo desiderato, effettuandone la rilocazione e il salvataggio: seguire per queste operazioni le istruzioni fornite all'inizio del programma stesso.

Doppia copia

di Damiano Bolla & Renzo Zonin

Visto il grande successo della routine di copy video presentata sul n. 13, abbiamo deciso di ... rifare le cose in grande. Presentiamo perciò una routine che esegue la copia dello schermo in doppia grandezza. La routine pilota le stampanti Epson e compatibili attraverso la porta seriale RS 232 dell'interface 1, e occupa 198 bytes; questa è la versione compattata di una routine che occupava inizialmente 370 bytes.

La riduzione di dimensioni è stata ottenuta mediante uno... sporco trucco di programmazione, noto agli addetti ai lavori come *codice non rientrante*, o automodificante.

La struttura è simile a quella dell'hard-copy in grandezza normale: c'è una sezione INIT che setta la stampante in modo grafico e richiama la subroutine TERZ; questa richiama UNTE, che stampa un terzo di schermo per volta, chiamando la routine LINH per stampare una semiriga (i 4 pixel piú alti o piú bassi di una linea), e le routines SHF1 e SHF2 per tenere in ordine i puntatori alla pagina video.

La LINH manda altri caratteri di controllo alla stampante e chiama CARH; questa si occupa di chiamare la routine CICL, che a sua volta scandisce fisicamente la mappa del video, e di modificare il codice della stessa CICL, in modo da farle testare successivamente il bit 7, 6, 5... 0 di ogni posizione (carattere) dello schermo.

La doppia grandezza è ottenuta facendo uscire un quadrato di 2 x 2 pixel per ogni pixel dello schermo.

Per chi volesse usare il codice

automodificante nei propri programmi, un avvertimento: questa tecnica è pericolosa! In caso di BREAK non potete essere sicuri dello stato in cui si trova il codice. In fase di stesura, un errore può alterare il programma in modo imprevedibile. Inoltre un programma scritto in questo modo non può essere eseguito in multitask (ma questo non preoccuperà gli utenti dello Spectrum).

Ora qualche nota pratica: la routine non è direttamente rilocabile, ma chi ha un assemblatore può spostarla semplicemente cambiando la ORG e il LOAD iniziali dello stato. Le versioni pubblicate sono per 48K (allocata dall'indirizzo 64000) e per 16K (posta alla locazione 32300). Ricordatevi di abbassare RAMTOP prima di caricare il codice, e salvate il codice oggetto ottenuto dando

SAVE "BIGCOPY" CODE inizio, inizio + 200.

Per attivare la routine, date RAND USR locazione di partenza; assicuratevi prima che la stampante sia collegata. Noi per le prove abbiamo usato la velocitá di 9600 baud.

Per finire un remake: ripubblichiamo la copy in grandezza normale, ma in versione compattata (171 bytes) e già rilocata, in modo da essere unita alla bigcopy. Potrete cosí avere in memoria un blocco compatto di soli 371 bytes, capace di eseguire copie grandi e piccole. Valgono le solite considerazioni su caricamento - uso - salvataggio; gli indirizzi di partenza sono 64200 (48K) e 32500 (16K).



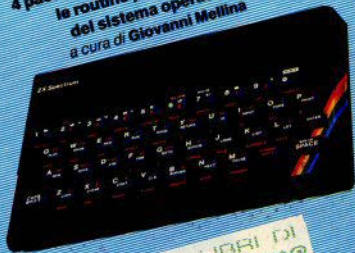
In edicola
questo mese

SINCLAIR
COMPUTER

PRESENTA

Tutti i segreti dello SPECTRUM

4 passi nella ROM: come utilizzare
le routine più importanti
del sistema operativo
a cura di Giovanni Mellina



I LIBRI DI
systems



SKOOL DAZE
Microsphere
Spectrum 48K

L'autore è S. Reidy, lo stesso che ha realizzato i più bei programmi per la Microsphere: Wheelie, Omnicalc 1 & 2, Train Game.

La scena è molto insolita: una scuola fornita di varie aule distribuite su due piani, con professori e alunni circolanti ognuno per i fatti propri, in una scenografia che occupa circa 8 volte lo schermo. Noi controlliamo uno studente che deve riuscire a rubare dalla cassaforte nell'ufficio del preside la propria pessima pagella. Per fare ciò deve però prima spaccare a colpi di fionde, con varie

tecniche di tiro, tutti i quadretti appesi sulle pareti della scuola, quindi estorcere la combinazione ai professori. L'insegnante di storia confesserà la cifra della combinazione della cassaforte soltanto quando verrà scritta sulla lavagna, per nostra mano la propria data di nascita. E per indovinarla bisogna assistere alle sue lezioni. Tutti i personaggi comunicano per mezzo di fumetti.

Quando spacciamo i quadri o siamo in giro durante le lezioni dobbiamo stare attenti a non farci sorprendere da qualche professore, pena tremende punizioni scritte, fino alla sospensione.

Oltre a tutto ciò, e al fatto che i professori vengono a cercarci se non siamo a lezione e qualcuno dei nostri compagni fa la spia (!), dobbiamo anche stare attenti a certi studenti. Per esempio c'è il bullo della scuola che tenta di metterci K.O. tutte le volte che ci ve-

de (e se un professore ci becca per terra ci punisce), e un certo "faccia d'angelo" che fa scherzi ai professori per poi far ricadere la colpa su di noi. Possiamo anche assegnare inizialmente a professori e compagni i nomi che vogliamo.

La trama sarebbe ancora più complessa, ma pensiamo di avere già detto abbastanza per invogliarvi all'acquisto di questo Skool Daze, certamente uno dei "softcapolavori" per Spectrum. Non per niente l'autore è un programmatore di potenti "mainframe", che a tempo perso scrive programmi in linguaggio macchina; a quanto si racconta, senza nemmeno usare un'assemblatore, ma battendo direttamente i codici binari dello Z80 nella memoria dello Spectrum!

Grafica: animata, eccellente. **9**
Comandi: ridefinibili, previsti joystick. **10**
Giocabilità: illimitata. **10**

dall'INGHILTERRA i fantastici computer games **ELETTTRIZZANTI**
MASTERTRONIC **AVVINCENTI**
EMOZIONANTI

4 NOVITÀ
OGNI MESE
del tuo rivenditore di fiducia.



THE CAPTIVE
CBM 64



ACTION BIKER
CBM 64 - SPECTRUM



SPECTIPEPE
COMMODORE '86



FINDERS KEEPERS
CBM 64 - MEX - SPECTRUM

TUFFATI NEL FANTASTICO MONDO MASTERTRONIC!
per vivere nuove emozionanti avventure piene di suspense e frenetiche animazioni.

QUALITÀ - PREZZO **solo £ 7900** è la grande proposta **MASTERTRONIC** per conquistare tanti amici.

I computer Sinclair sono dotati all'origine di un modulatore UHF, che fornisce un segnale di tipo "televivo", adatto cioè ad essere inviato a un normale televisore domestico (la modulazione del segnale è quella che consente la sua trasmissione attraverso l'aria); ciò è un indubbio vantaggio per un apparecchio economico come lo Spectrum (o il vecchio ZX81), poiché consente a tutti di utilizzare il computer, senza dover acquistare il monitor.

Un monitor può essere definito, con un'espressione un po' drastica, un televisore monco: è privo, infatti, della sezione di demodulazione, e non può ricevere un segnale via etere, dall'antenna. Già questo fatto, di non dover attraversare due trasformazioni (modulazione e demodulazione), consente però un guadagno nella qualità dell'immagine; inoltre i monitor sono dotati, solitamente, di tubi catodici migliori, per una più alta definizione.

Se avete lo Spectrum collegato a un normale televisore, avrete notato che, dopo un certo tempo dall'accensione, è spesso necessario "ristimare" l'immagine, agendo sulla sintonia del tv: il funzionamento prolungato provoca delle piccole variazioni nel segnale in uscita; la causa principale è il fisiologico riscaldamento, che modifica, tra l'altro, la sequenza del clock al quarzo, le cui oscillazioni scandiscono l'esecuzione di tutto quello che avviene sulla scheda.

Altri disturbi all'immagine possono provenire dall'esterno: emittenti private su canali UHF molto vicini al 36 (su cui "trasmette" il computer), elettrodomestici privi di schermi antidisturbo, la semplice posizione del filo che collega Spectrum e tv, sono fattori possibili del peggioramento dell'immagine.

Se lavorate davanti al video per molte ore, soprattutto con word processor o spreadsheet (quindi con testi, piuttosto che immagini), una visualizzazione instabile e poco nitida diviene presto insopportabile, e il desiderio di un buon monitor sarà più che giustificato.

Purtroppo lo Spectrum non possiede l'uscita per il collegamento a un monitor, ma è possibile rimediare

Intorno al monitor

di Fabrizio Ferrario

una senza grosse difficoltà.

Questa uscita monitor può essere realizzata in diversi modi: quello che proponiamo qui è sicuro ed economico. Il sistema più semplice sarebbe quello di collegare direttamente l'ingresso monitor al segnale non modulato, a monte cioè del modulatore dello Spectrum. La soluzione è già stata da noi applicata con successo a diversi esemplari, ma deve essere usata solo con monitor in perfette condizioni: c'è infatti il rischio che un ritorno di tensione da parte del monitor, evento per fortuna non frequente, provochi danni allo Spectrum (il più grave potrebbe essere la bruciatura del chip ULA).

Con il sistema che suggeriamo qui, il computer sarà al sicuro da qualsiasi pericolo: l'adattatore di

uscita ha anche un trimmer per poter regolare il livello del segnale, allo scopo di non saturare particolari monitor con ingressi molto sensibili.

C'è ancora da fare una precisazione: non tutti i monitor si possono collegare allo Spectrum, ma solamente quelli aventi una presa video PAL-composito; quindi quelli dotati di ingresso RGB o TTL, o altri custom ancora, *non sono collegabili*.

Da un po' di tempo a questa parte i fabbricanti di televisori, grazie ad accordi a livello CEE, forniscono i loro apparecchi di una presa monitor; potete imbattervi in due tipi di connettore: AV o SCART (ultimamente si sta generalizzando l'impiego del secondo).

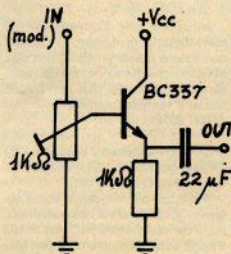
La prima è una presa DIN a 6 poli, mentre l'altra è una presa europea da 21 poli, simile a quelle a 25 poli dei connettori per interfaccia RS-232. Attraverso queste prese, possiamo utilizzare il televisore come monitor: per la AV si devono collegare VIDEO e GND (ground, cioè massa), mentre per la SCART si collegano il pin 20 (segnale) e il pin 17 (massa).

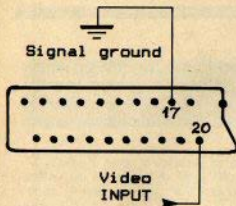
Per vedere l'immagine del computer dovrete commutare l'apparecchio televisivo per la ricezione dalla presa monitor; nel modo indicato sulle istruzioni: o c'è un apposito interruttore, o si agisce con il telecomando.

Ma veniamo al circuito, che, diciamo per i più esperti, non è altro che un inseguitore di emettitore con il carico disaccoppiato.

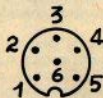
Può essere montato su una ba-

Il collegamento al monitor





connettore SCART



connettore DIN

setta o, data l'estrema semplicità, essere lasciato "volante": l'importante è che la costruzione sia eseguita a regola d'arte, da qualcuno con una buona pratica di elettronica: se non è il vostro caso, troverete sicuramente un amico che vi dia una mano.

Il circuito va collegato a monte del modulatore dello Spectrum, al filo che porta il segnale (è isolato da una piastrina di plastica), mentre l'alimentazione si ricava dall'altro filo che entra nello scatolino e che è collegato a un condensatore passante, saldato sullo scatolino stesso.

Il cavo per l'uscita monitor deve essere coassiale, possibilmente avente un'impedenza di circa 50 ohm.

Tutto quanto detto vale anche per lo ZX80 e ZX81; il QL, fortunatamente, dispone sia di un'uscita modulata, per televisore, che di uscita monitor, e questa dispone a sua volta sia di segnale composito che RGB: la piedinatura del connettore DIN è riportata a pag. 31 della sezione *Concepts* del manuale.

Un tasto per volta

B

NORMALE

Comando BORDER. Modifica il colore della cornice attorno alla finestra video (cioè dell'area su cui non si può scrivere). I valori consentiti sono da 0 a 7.

SYMBOL SHIFT

Operatore "*" (asterisco). È il segno di moltiplicazione, di uso evidente, per qualsiasi operazione algebrica. Serve inoltre come identificatore dopo LOAD, SAVE, VERIFY, MERGE, quando si vuole operare con i microdrives anziché con il registratore.

CAPS SHIFT

"B" minuscola.

MODO "E"

Istruzione BIN. Converte un numero binario in numero decimale, con valore massimo 65535 (16 bit); essendo una funzione, non può essere usata da sola, cioè non potete scrivere semplicemente
BIN 10100111

Deve invece essere preceduta da un'assegnazione: LET variabile = BIN ...; PRINT BIN...; etc.

Il suo uso più frequente, come abbiamo visto in altre occasioni (cfr. l'uso del tasto GRAPHIC, sul n. 10), è nella costruzione dei caratteri grafici definiti dall'utente: una volta di-

segnata la maschera del carattere, si assegna 1 ai pixel inchiostrati e 0 agli altri. I numeri binari che si ottengono, leggendo il carattere per righe orizzontali, possono essere usati, preceduti da BIN, risparmiandosi di calcolare i decimali: l'esempio riportato, che definisce il carattere grafico assegnato al tasto A, chiarirà meglio il procedimento.

```
10 FOR i=0 TO 7
20 READ char
30 POKE USR "a"+i, char
40 NEXT i
50 DATA BIN 00111100, BIN
01111110, BIN 11111111, BIN
01111110, BIN 00111100, BIN
00011000, BIN 00111100, BIN
11111111
```

Questo programma monoriga esegue invece la conversione decimale/binario, per numeri fino a 8 bit (POKE in 16384 del numero da convertire):

```
LET a$="": FOR f=0 TO 7: LET
a$=a$+ STR$ POINT (f, 175):
NEXT f: PRINT a$
```

SYMBOL S. in "E"

Comando BRIGHT. Luminosità dell'area di scrittura: conferisce al campo in cui si sta scrivendo una luminosità maggiore rispetto alla parte circostante; si attiva con argomento 1 e si disattiva con 0; accetta inoltre argomento 8 (trasparenza).

Se eseguito come istruzione a stante, la maggiore luminosità riguarderà l'intera finestra video, ma



si può limitare l'effetto a una stringa, fino a un singolo carattere, associandolo a PRINT o a PLOT:

```
PRINT BRIGHT 1; "stringa"  
PLOT BRIGHT 1;x,y
```

N

NORMALE

Istruzione NEXT. Istruzione di cui si è già ampiamente detto trattando il FOR, cui è associata binuovocamente: questa non può sussistere senza quella, e viceversa. A differenza di quanto avviene nel basic Microsoft, con lo Spectrum è obbligatorio specificare, dopo il NEXT, la variabile di ciclo associata. Cosa peraltro sempre opportuna, nel caso di *nesting* (più cicli nidificati, cioè inseriti uno nell'altro), onde evitare confusioni.

SYMBOL SHIFT

Operatore "," (virgola). Serve come separatore in molte situazioni: con PRINT provoca un salto in avanti della posizione corrente di stampa; se questa è tra la colonna 0 e la 15 si va alla col. 16 della stessa riga, se è oltre la 15esima si salta alla prima colonna della riga successiva.

Separa un dato dall'altro nelle linee DATA; le coordinate negli enunciati PLOT, DRAW, CIRCLE, AT; gli indici negli array pluridimensionali.

Ha una funzione sintattica con OPEN, CLOSE, etc., relativamente ai canali "l", "b", "n". Eccezioni.

L'unico caso in cui non serve proprio è come separatore tra unità e decimali: qui si deve sempre usare il punto.

CAPS SHIFT

"N" maiuscola.

MODO "E"

Istruzione INKEY\$. Normalmente serve per leggere dalla tastiera: PRINT INKEY\$ stampa il carattere corrispondente al tasto premuto. È

utile perciò in un menu di opzioni, identificabili ciascuna con una lettera o un numero:

```
IF INKEY$ = "x" THEN GO TO...
```

La differenza rispetto a INPUT è che accetta solo un carattere per volta e non serve dare ENTER, il che, nel caso appunto di un menu, rappresenta un notevole vantaggio pratico.

Le possibilità di INKEY\$ sono però molto ampliate dalla presenza della ROM-ombra dell'interfaccia 1: ai pari di INPUT, è possibile assegnare questa funzione a un canale logico, che a sua volta sarà collegato a un *device* esterno: microdrive, network, RS-232.

Nel primo caso, INKEY\$ preleva un carattere per volta da files DATA, nel secondo da un altro Spectrum, nel terzo da qualsiasi dispositivo che glieli possa inviare serialmente (tipicamente, un altro computer, eventualmente attraverso modem).

La procedura è sempre quella: occorre tenere "in attesa" il programma finché non arriva un carattere, con un'istruzione tipo

```
10 IF INKEY$ "" THEN GO TO 10
```

Specificando il canale d'ingresso, se non è dalla tastiera; l'esecuzione prosegue quando arriva qualcosa:

```
20 IF INKEY$ <"> THEN...
```

Avrete compreso che questa è una delle istruzioni più importanti per poter dialogare con il computer: consigliamo a tutti, particolarmente ai possessori di un'interfaccia RS-232, di fare molti esperimenti. Che funzione usa lo Spectrum per rispondere ai vostri INPUT o INKEY\$?

SYMBOL S. in "E"

Comando OVER. Un altro comando per lo schermo, che esegue le sovrimpressioni: dopo aver dato OVER 1, se scrivete qualcosa in una posizione in cui già compaiono pixel "inchiostriati", questi non vengono cancellati, come solitamente avviene, ma rimangono.

Più precisamente, rimangono quelli che non coincidono con i nuovi: viene infatti eseguito un XOR logico (un OR esclusivo) sui pixel inchiostriati.

OVER si usa soprattutto con DRAW, PLOT, CIRCLE, e in genere per disegnare sullo schermo, poiché consente di aggiungere parti nuove al disegno senza mai perdere il precedente.

M

NORMALE

Istruzione PAUSE. Sospende l'esecuzione del programma basic per un periodo controllabile con l'argomento (numerico e obbligatorio). Il range utile è da 0 a 65535, e scandisce 50esimi di secondo: con il valore massimo l'interruzione vale 21'50", con 0 la sospensione prosegue indefinitamente, finché non si preme un tasto; è utile perciò nei casi in cui il programma deve attendere un intervento dell'utente, senza un tempo prestabilito: per esempio, quando si mostrano pagine di istruzioni.

SYMBOL SHIFT

Operatore "." (punto). Separatore tra le unità e i decimali.

CAPS SHIFT

"M" maiuscola.

MODO "E"

Operatore PI (*pi greco*). Inserisce nel calcolo un valore di *pi-greco* approssimato alla settima cifra decimale.

SYMBOL S. in "E"

Comando INVERSE. Ultimo dei comandi destinati al controllo dello schermo, serve per "invertire" appunto, la matrice di punti stampata.

Gli argomenti sono ancora 1 e 0, per attivare e disattivare; se si esegue come statement autonomo, lo scambio di colore tra carta e inchiostro riguarda l'intero display-file; associato a una qualsiasi delle istruzioni che scrivono sul video, limita il suo effetto a quel comando.

Utilizzate
i tagliandi
per abbonarvi,
collaborare,
chiedere o
darci consigli,
pubblicare
un'inserzione
per comprare,
vendere,
cercare
contatti.

Ritagliate
e spedite
in busta
chiusa a:
Sinclair
Computer,
viale
Famagosta 75,
20142 Milano.

Registrate il mio abbonamento annuale a:

Sinclair Computer (L. 28.000) Computer + Sinclair Computer (L. 55.000)

Ho versato l'importo sul c/c postale n. 37952207

Accludo assegno non trasferibile n. _____

banca _____

intestato a SYSTEMS Editoriale, v.le Famagosta 75, 20142 Milano

Il mio computer è ZX81 Spectrum

possiedo SI NO

Nome _____

stampante

Via _____

n. _____

microdrive

CAP. [] [] [] [] [] città _____

joystick

Tel. _____

prov. _____

N.B. L'abbonamento è annuale (11 numeri) e decorre dal primo numero edito dopo il ricevimento della sottoscrizione.

ABBONAMENTO

Desidero collaborare a Sinclair Computer

COLLABORAZIONE

Invio il programma "

listato e registrato su cassetta, con un articolo di commento.

Garantisco che il software è originale e vi autorizzo a pubblicarlo.

Per il compenso scrivete mi il seguente indirizzo:

Nome _____

Via _____

n. _____

CAP. [] [] [] [] [] città _____

prov. _____

Tel. _____

N.B. Il materiale anche non pubblicato non viene restituito

HELP

Nome _____

Via _____

n. _____

CAP. [] [] [] [] [] città _____

Tel. _____

prov. _____

Questo mese ho acquistato / provato i seguenti programmi
e li valuto così (max tre titoli):

CLASSIFICA

TITOLO	Ottimo	Buono	Mediocre	Deludente

nome e indirizzo (facoltativo) _____

VENDIC

COMPRO

INSERZIONI

Avvertenze:
ABBONAMENTI: scrivete l'indirizzo completo

COLLABORAZIONE: il listato non è indispensabile, la cassetta sì.

HELP: non accludete francobolli, non rispondiamo privatamente

INSERZIONI: la rubrica è destinata agli scambi tra privati; la redazione si riserva il diritto di modificare o cestinare gli annunci palesemente speculativi.

Nome _____

Via _____

n. _____

CAP. [] [] [] [] [] città _____

Tel. _____

prov. _____



Un'iniziativa condotta con la nota rivista Computer



PROGRAMMO IN BASIC

Il linguaggio del futuro in un manuale rapido e completo di Clizio Merli
 pagg. 224 (L. 9.000)

Il Basic, attualmente il linguaggio più conosciuto - adatto all'utilizzo su qualunque tipo di macchina e in particolare sul personal e gli home-computer - può essere appreso in poche ore con l'ausilio di questo agile manuale.



COME SCEGLIERE UN COMPUTER

Guida pratica per l'acquisto di un mini o di un micro computer professionale di Michele Di Pisa
 pagg. 160 (L. 6.000)

Quale modello scegliere tra gli oltre 600 computer commercializzati in Italia? La conoscenza delle caratteristiche delle varie macchine è indispensabile. Con un approccio a "menu" l'Autore vuol essere guida proprio in questa fase.



UTILITY E ROUTINE PER IL COMMODORE 64

di Gloriano Rossi
 pagg. 192 (L. 9.000)

L'esecuzione di una istruzione BASIC può richiedere diverse centinaia di passi di programmi in linguaggio macchina. La dimensione dei programmi è ciò che intimidisce maggiormente l'utilizzatore medio di Commodore: aiutato da questo testo chiunque potrà affrontare senza problemi il processo di scrittura di un programma.



BASIC PER LO SPECTRUM

di Maurizio Ariens e Clizio Merli
 pagg. 192 (L. 9.000)

Un libro per quanti hanno acquistato il computer ZX Spectrum della Sinclair e intendono sfruttarne appieno tutte le capacità, dall'hardware alla programmazione in assembly (linguaggio macchina).

I volumi, che sono comunque in vendita nelle migliori librerie di tutta Italia, possono anche essere richiesti direttamente all'Editore.
 Importante: l'ordine minimo dovrà essere di L. 15.000.



Edizioni ACANTHUS

VIALE GRAN SASSO, 23 - 20131 MILANO

Inviatemi i seguenti volumi:

Titolo	quantità	prezzo unitario
spese postali		L. 2.000
totale		L.

Pagherò contrassegno il dovuto (più L. 2.000 per contributo spese postali) al ricevimento. Potrà restituire i libri entro 8 giorni se non saranno di mio gradimento e avere il rimborso immediato.

COGNOME _____

NOME _____

VIA _____ N. _____

C.A.P. _____ CITTÀ _____

FIRMA _____

DATA _____

Scrivere in stampatello e spedire in busta chiusa.

Alberto Cultrera,
 Monica Di Pisa, Daniela Giacomelli

DIZIONARIO DELL'INFORMATICA

Vocabolario Inglese-Italiano
 di Cultrera, Di Pisa, Giacomelli
 pagg. 388 (L. 25.000)

Uno strumento indispensabile per chi si avvicina al mondo dell'informatica e per gli specialisti che hanno l'esigenza di accedere alla dinamica letteratura anglosassone.

Computer



La torre di Hanoi

(C) 1985 by Grandi Fabio

```

0>REM +-----+
+-----+
|      |      |      |      |
+-----+
> ? 1985 by Grandi F
abio < |      |      |      |
|      |      |      |      |
+-----+
100 LET HANOI=180: LET OUTPUT=2
80
110 LET push=310: LET pop=370
120 LET c=CODE "A": LET d=c+1:
LET e=d+1
130 INPUT "Numero di dischi? ";
b: IF b<=1 THEN GO TO 420
140 PRINT AT 0,0;"Il disco da
l piolo al piolo": DIM s(4*(b
-1)): LET sp=0
150 GO SUB HANOI: REM Hanoi (N,X
,Y,Z)
160 PRINT #0;"premi un tasto pe
r continuare": PAUSE 0: RUN
170 REM HANOI
180 IF b=1 THEN GO TO OUTPUT:
REM mosca "X 1 Z"
190 GO SUB push: LET b=b-1
200 LET a=e: LET e=d: LET d=a
210 GO SUB HANOI: REM Hanoi (N-1
,X,Z,Y)
220 GO SUB pop: GO SUB OUTPUT:
REM mosca "X N Z"
230 GO SUB push: LET b=b-1
240 LET a=c: LET c=d: LET d=a
250 GO SUB HANOI: REM Hanoi (N-1
,Y,X,Z)
260 GO SUB pop: RETURN
270 REM OUTPUT
280 POKE 23692,255: PRINT TAB 7
;b;TAB 17;CHR# c;TAB 28;CHR# e
290 RETURN
300 REM push
310 LET sp=sp+1: LET s(sp)=b
320 LET sp=sp+1: LET s(sp)=c
330 LET sp=sp+1: LET s(sp)=d
340 LET sp=sp+1: LET s(sp)=e
350 RETURN
360 REM pop
370 LET e=s(sp): LET sp=sp-1
380 LET d=s(sp): LET sp=sp-1
390 LET c=s(sp): LET sp=sp-1
400 LET b=s(sp): LET sp=sp-1
410 RETURN
420 LIST
  
```

```

10 BORDER 1: PAPER 0: INK 5
20 REM *** HEX-LOADER ***
30 CLEAR 31999: READ A,B,C,D,E
,F: DATA 10,11,12,13,14,15
40 LET c#= "CD6B0D3E02CD0116060
00E4116421E433E01B8281AC5D5057B5
A57CD107DD1C1CD2F7DC5D505794A57C
D107DD1C1C93E17D73E0DD70779D73E2
0D778FE0A38063E31D0778D06AC630D73
E20D778D73E0DD7C9"
50 FOR i=1 TO LEN c#/2
60 POKE 31999+i,16*VAL c$(2*i-
1)+VAL c$(2*i): NEXT i: CLEAR
70 REM *** FINE HEX-L ***
80 PRINT PAPER 2: INK 7;"
  
```

UZIONE RICORSIVA del GIOCO: SOL

"LA TORRE DI HANOI"

```

90 PRINT "" Significato dell
e mosse:""" " : INVERSE 1;" X
m Y "; INVERSE 0;" = muovi il di
sco numero m da
l palo ""X"" al pa
lo ""Y""
100 PRINT AT 19,4: PAPER 4;"
  
```

```

110 INK 7: FOR i=1 TO 3: PLOT 6
5*i+12,24: DRAW 0,50: DRAW 6,0:
DRAW 0,-50: NEXT i: INK 5
120 PRINT INK 7;" palo A
palo B palo C"
130 PRINT AT 13,2;"disco:"; INK
6;AT 15,2;"1 -";AT 16,2;"2 -";A
T 17,2;"...";AT 18,2;"N -"
140 LET w$=CHR# 135: LET o$=CHR
# 139: LET u$=CHR# 131+CHR# 131:
PRINT PAPER 3: INK 6;AT 15,8;o
$;u$;w$;AT 16,7;o$;u$;u$;w$;AT 1
8,6;o$;u$;u$;u$;w$;
150 INPUT " Numero di dischi [1
/19]? ";n: IF n<1 OR n>19 THEN
GO TO 150
160 POKE 32009,n: INK 4: RANDOM
IZE USR 32000
170 PRINT #0;"Premi un tasto pe
r continuare": PAUSE 0: INPUT ""
: INK 5: RUN 80
  
```

7D00	CD6B0D	CALL	#0D6B
7D03	3E02	LD	A,#02
7D05	CD0116	CALL	#1601
7D08	0605	LD	B,#05
7D0A	0E41	LD	C,#41
7D0C	1642	LD	D,#42
7D0E	1E43	LD	E,#43
7D10	3E01	L7D10 LD	A,#01

0>REM

LISTA

```

7D12 B8 CP B
7D13 281A JR Z,L7D2F
7D15 C9 PUSH BC
7D16 D5 PUSH DE
7D17 05 DEC B
7D18 7B LD A,E
7D19 5A LD A,D
7D1A 57 LD D,A
7D1B CD107D CALL L7D10
7D1E D1 POP DE
7D1F C1 POP BC
7D20 CD2F7D CALL L7D2F
7D23 C5 PUSH BC
7D24 D5 PUSH DE
7D25 05 DEC B
7D26 79 LD A,C
7D27 4A LD C,D
7D28 57 LD D,A
7D29 CD107D CALL L7D10
7D2C D1 POP DE
7D2D C1 POP BC
7D2E C9 RET
7D2F 3E17 L7D2F LD A,#17
7D31 D7 RST #10
7D32 3E0D LD A,#0D
7D34 D7 RST #10
7D35 D7 RST #10
7D36 79 LD A,C
7D37 D7 RST #10
7D38 3E20 LD A,#20
7D3A D7 RST #10
7D3B 7B LD A,B
7D3C FE0A CR #0A
7D3E 3806 JP C,L7D46
7D40 3E31 LD A,#31
7D42 D7 RST #10
7D43 7B LD A,B
7D44 D60A SUB #0A
7D46 C630 L7D46 ADD A,#30
7D48 D7 RST #10
7D49 3E20 LD A,#20
7D4B D7 RST #10
7D4C 7B LD A,E
7D4D D7 RST #10
7D4E 3E0D LD A,#0D
7D50 D7 RST #10
7D51 C9 RET

```

```

7 LET ns=VAL n$: IF ns=0 THEN
GO TO 5
8 FOR d=1 TO ns
9 DIM x(32): DIM a(8): DIM c(
5): DIM g(5): DIM s(5): DIM w(5)
10 LET scr=0: LET cr=0: LET f1
=0: LET k=0: LET i=0: LET s$=""
11 LET y=8: LET yy=103: LET pt
=0
12 IF d>1 THEN CLS : FOR f=8
TO 13: PRINT AT f,4: PAPER 5: FL
ASH 1: "
: NEXT f
13 PRINT AT 9,10: PAPER 6: BRI
GHT 1: " P O K E R ! "; AT 12,4: " A
TTENDI,STO MESCOLOANDO "
14 RESTORE : GO SUB 9000
15 GO SUB 5000: CLS
16 REM *****
18 REM * POKER *
20 REM *****
30 FOR f=1 TO 5
40 LET c(f)=VAL (STR$ x(f)) (2
TO ): LET s(f)=VAL (STR$ x(f)) (1
)
45 LET a(c(f)-6)=a(c(f)-6)+1
50 NEXT f: GO SUB 7500: GO SUB
9500: GO SUB 8500: GO SUB 7000:
GO SUB 500
56 PRINT #0: BRIGHT i: INVERSE
i: "A--->accetto", "L--->lascio",
: PAUSE 0
57 IF INKEY$("<">"a" AND INKEY$("<">
"1" THEN GO TO 57
60 IF INKEY$="1" THEN LET sg=
sg-1000: PRINT AT 2,22: FLASH 1:
" LASCIO! ": INPUT #0: GO TO 370
70 LET sg=sg-ap: LET pt=pt+ap:
PRINT AT 10,22: " PIATTO: "; AT 1
1,22: " $ "; " " AND pt<1000:pt;"
" AND pt<10000;" "
80 INPUT INVERSE 1: BRIGHT 1:
"Scarto: "; LINE c$
90 IF LEN c$>4 THEN GO TO 80
95 LET cs=LEN c$
100 FOR f=1 TO cs
105 IF CODE c$(f)<49 OR CODE c$
(f)>53 THEN GO TO 80
115 LET xc=VAL c$(f)
120 LET c(xc)=VAL (STR$ x(10+f)
) (2 TO ): LET s(xc)=VAL (STR$ x(
10+f)) (1)
130 NEXT f
140 GO SUB 8500: GO SUB 7000
141 DIM a(8): FOR f=1 TO 5
142 LET a(c(f)-6)=a(c(f)-6)+1
143 NEXT f
150 GO SUB 1000: LET h1=f1
160 FOR f=1 TO 5
170 LET g(f)=c(f): LET w(f)=s(f
): LET gr=cr
175 NEXT f
180 LET f1=0: DIM a(8)

```

Poker

```

1 BORDER 1: PAPER 7: INK 0: C
LS : RANDOMIZE PEEK 23672: LET s
g=50000
2 FOR f=8 TO 13: PRINT AT f,4
: PAPER 5: FLASH 1: "
: NEXT f
3 PRINT AT 9,10: PAPER 6: BRI
GHT 1: " P O K E R ! "
5 INPUT INVERSE 1: BRIGHT 1:
"Numero smazzate?(1/99) "; LINE
n$
6 IF LEN n$>2 OR CODE n$(1)<4
8 OR CODE n$(1)>57 OR CODE n$(LE
N n$)<48 OR CODE n$(LEN n$)>57 T
HEN GO TO 2

```



ilistati

```
200 REM gioco del computer
210 FOR f=1 TO 5
220 LET c(f)=VAL (STR$ x(5+f))
2 TO ): LET s(f)=VAL (STR$ x(5+f))
) (1)
230 LET a(c(f)-6)=a(c(f)-6)+1
240 NEXT f
241 GO SUB 1000
245 LET yy=15: LET y=19: LET i=
11+cs
246 IF f1=5.5 OR f1=8 OR f1=10
THEN LET scr=0: GO TO 280
247 IF f1=1 THEN LET scr=4: GO
SUB 2500: GO TO 280
248 IF f1=2 THEN GO SUB 8700
250 FOR f=1 TO 8
260 IF a(f)=1 THEN LET scr=scr
+1: GO SUB 2000
270 NEXT f
280 PRINT AT 13,22: FLASH 1;"SC
ARTO: ";scr
282 IF k=1 THEN GO TO 350
285 DIM a(8): LET cr=0: LET f1=
0
290 FOR f=1 TO 5
300 LET a(c(f)-6)=a(c(f)-6)+1
310 NEXT f
320 GO SUB 1000: GO SUB 600
330 GO SUB 8500: GO SUB 7000
340 IF f1>h1 THEN PRINT AT 15,
22: FLASH 1;"VITTORIA!": GO TO 3
60
345 IF f1=h1 AND cr>=gr THEN P
RINT AT 15,22: FLASH 1;"VITTORIA
!": GO TO 360
350 PRINT AT 8,22: FLASH 1;"VIT
TORIA!": LET sg=sg+pt
360 IF sg<0 THEN PRINT #0; BRI
GHT 1; FLASH 1;"HAI ESTINTO I TU
OI FONDI!": PAUSE 300: GO TO 420
365 IF sg>100000 THEN INPUT #0
: PRINT #0; INVERSE 1; BRIGHT 1;
FLASH 1;"HAI SBANCATO IL CASINO
!": PAUSE 300: GO TO 420
370 PRINT #0; INVERSE 1; FLASH
1; BRIGHT 1; TAB 1;"DISPONI DI: $
";sg; ": PAUSE 300
380 REM *****
390 NEXT d: REM ciclo di gioco
400 REM *****
410 INPUT #0: PRINT #0; AT 0,0;
BRIGHT 1; PAPER 2; " LA PART
ITA E' FINITA " : PAUSE 25
413 IF sg<50000 THEN PRINT #0;
BRIGHT 1; PAPER 1; FLASH 1; "
HAI VINTO TU " :
PAUSE 0
417 IF sg<=50000 THEN PRINT #0
; BRIGHT 1; PAPER 1; FLASH 1; "
HO VINTO IO! " :
PAUSE 0
420 INPUT INVERSE 1; BRIGHT 1;
"Fai un'altra partita?(s/n) "; L
```

```
INE v$
430 IF v$("<"s" AND v$("<"n" THEN
GO TO 420
440 IF v$="s" THEN RUN
450 STOP
499 REM SUBR per le puntate
500 LET ap=(INT (RND*9)+1)*100
505 LET pt=pt+ap
510 PRINT AT 10,22;"APERTURA: ";
AT 11,22;" $ ";ap; " "
520 RETURN
599 REM II SUBR per puntare
600 IF LEN (STR$ (d/2))>LEN (ST
R$ INT (d/2)) THEN GO SUB 8800:
LET k=1: GO SUB 601: LET k=0: R
ETURN
601 INPUT #0; AT 0,0; BRIGHT 1;
PAPER 2;"Puntata min.: $1000 max
.: $10000"; PAPER 7; INK 0;"Punt
ata(0=passo): " AND d/2=INT (d/2)
;"Rilancio(0=passo): " AND d/2>I
NT (d/2); LINE p$
602 FOR f=1 TO LEN p$
603 IF CODE p$(f)<48 OR CODE p$
(f)>57 THEN GO TO 600
604 NEXT f: LET pg=VAL p$
605 IF pg=0 AND k=1 THEN RETUR
N
606 IF pg=0 THEN PRINT AT 2,22
; FLASH 1;" LASCIO! " : GO TO 360
610 IF pg<1000 OR pg>10000 THEN
GO TO 600
620 PRINT AT 2,22;"RILANCIO: ";A
T 3,22;" $ ";pg; " " " AND pg<1
0000
630 LET sg=sg-pg: LET pt=pt+pg
640 REM Decisione del computer
650 IF f1=2 AND cs<3 OR f1=2 AND
D pg>5000 OR f1=2 AND cs>=3 AND
cr<13 THEN PRINT AT 19,22; FLAS
H 1;" LASCIO! " : LET sg=sg+pt: G
O TO 360
651 IF cs=0 AND f1<5.5 THEN PR
INT AT 19,22; FLASH 1;" LASCIO!
": LET sg=sg+pt: GO TO 360
652 IF f1=4 AND cr<13 AND cs=1
AND pg>=5000 THEN PRINT AT 19,2
2; FLASH 1;" LASCIO! " : GO TO 36
0
653 IF f1=1 THEN PRINT AT 19,2
2; FLASH 1;" LASCIO! " : LET sg=s
g+pt: GO TO 360
655 IF f1<5AND INT (RND*10)=0AND
D pg>1000OR f1<5AND scr=4AND cs<
=3AND pg>4000OR f1<5AND pg>=8000
THEN PRINT AT 19,22;" LASCIO! " :
LET sg=sg+pt:GO TO 360
656 IF f1=5 AND cr<11 AND pg>70
00 AND (cs=2 OR cs=3) THEN PRIN
T AT 19,22; FLASH 1;" LASCIO! " :
LET sg=sg+pt: GO TO 360
660 PRINT AT 19,22;" CI STO! " ;
AT 2,22; PAPER 0;" " " ; AT
```

```

3,22;"      ": LET pt=pt+pg
665 PRINT AT 11,25;pt: PAUSE 10
0
670 IF k=0 THEN LET k=1: GO SU
B 8800: LET k=0
680 RETURN
1000 REM scala
1010 FOR h=1 TO 4
1020 FOR f=0 TO 4
1030 IF a(f+h)<>1 THEN GO TO 10
60
1040 NEXT f
1050 IF f=5 THEN LET fl=5.5: LE
T cr=10+h: RETURN
1060 NEXT h
1100 REM coppia,doppia coppia,tr
is,poker
1110 FOR f=1 TO 8
1120 IF a(f)=4 THEN LET fl=9: L
ET cr=f+6: RETURN
1130 IF a(f)=3 THEN LET fl=f+5
: LET cr=f+6
1140 IF a(f)=2 THEN LET fl=f+2
: IF cr=0 OR (fl=4 AND cr<f+6) T
HEN LET cr=f+6
1150 NEXT f
1160 IF fl>0 THEN RETURN
1200 REM colore,scala reale
1210 LET f=1
1220 IF s(f)=s(f+1) AND s(f)=s(f
+2) AND s(f)=s(f+3) AND s(f)=s(f
+4) THEN LET fl=8+(2 AND fl=5):
RETURN
1230 LET fl=1
1240 FOR f=8 TO 1 STEP -1
1250 IF a(f)=1 THEN LET cr=f+6:
GO TO 1270
1260 NEXT f
1270 RETURN
1999 REM SUBR per lo scarto
2000 FOR z=1 TO 5
2010 IF c(z)=f+6 THEN LET c(z)=
VAL (STR$ x(i))(2 TO ): LET s(z)
=VAL (STR$ x(i))(1): GO TO 2030
2020 NEXT z
2030 LET i=i+1: RETURN
2499 REM II SUBR per lo scarto
2500 REM SUBR per lo scarto (car
ta isolata o scala aperta)
2510 FOR h=1 TO 4
2520 FOR f=0 TO 3
2530 IF a(f+h)<>1 THEN GO TO 25
60
2540 NEXT f
2550 IF f=4 THEN LET a(h)=0: LE
T a(h+1)=0: LET a(h+2)=0: LET a(
h+3)=0: LET scr=0: GO TO 250
2560 NEXT h
2570 FOR f=2 TO 5
2580 LET c(f)=VAL (STR$ x(i))(2
TO ): LET s(f)=VAL (STR$ x(i))(1
)
2590 LET i=i+1: NEXT f
2600 RETURN
5000 REM SUBR per il mescolament
o delle carte
5010 LET u=0
5020 FOR h=1 TO 4
5030 FOR f=7 TO 14
5040 LET u=u+1
5050 LET x(u)=f+h*100
5060 LET f: NEXT h
5070 FOR f=1 TO 16
5080 LET nc=INT (RND*32)+1
5090 LET xx=x(f): LET x(f)=x(nc)
: LET x(nc)=xx
5100 NEXT f: RETURN
7000 REM SUBR per la visualizzaz
ione delle carte
7005 PLOT 0,87: DRAW 167,0: PLOT
167,0: DRAW 0,175
7010 PLOT 15,yy: DRAW 137,0: DRA
W 0,57: DRAW -137,0: DRAW 0,-57
7020 FOR f=39 TO 111 STEP 24
7030 PLOT f,yy: DRAW 0,56
7040 NEXT f
7041 IF k=1 THEN RETURN
7042 FOR f=1 TO 5
7044 PRINT AT y+1,(f*3)+1: INVER
SE 1;f
7046 NEXT f
7047 FOR f=0 TO 4
7053 LET s$=""
7060 LET s$=s$+"J" AND c(f+1)=1
1)+("Q" AND c(f+1)=12)+("K" AND
c(f+1)=13)+("A" AND c(f+1)=14)+
STR$ c(f+1) AND c(f+1)<11)
7070 RESTORE 9989+s(f+1): READ e
$
7080 PRINT AT y,2+f*3: BRIGHT 1:
INK 0+(2 AND s(f+1)<3):s$:AT y-
6,2+f*3:s$:AT y-3,4+f*3: OVER 1:
e$
7090 NEXT f
7100 PRINT AT y,18-(1 AND LEN s$
=2): BRIGHT 1: INK 0+(2 AND s(f)
<3):s$:AT y-6,18-(1 AND LEN s$=2
):s$
7110 RETURN
7499 REM SUBR per la visualizzaz
ione delle carte coperte
7500 FOR f=13 TO 19
7510 PRINT AT f,2: PAPER 4;"
"
7520 NEXT f
7530 LET k=1: LET yy=15: GO SUB
7000: LET k=0: LET yy=103: RETURN
8500 REM **SUBR per cancellare**
8510 FOR b=y TO y-6 STEP -1
8520 PRINT AT b,2: BRIGHT 1;"
"
8530 NEXT b
8540 RETURN
8550 REM *****
8699 REM SUBR: scarto mascherato

```

```

8700 FOR f=1 TO 5
8710 IF a(c(f)-6)=1 THEN GO TO
8730
8720 NEXT f
8730 IF INT (RND*3)=0 THEN LET
a(c(f)-6)=0
8740 RETURN
8750 REM *****
8799 REM III SUBR per puntare
8800 IF f1<5 AND k=1 THEN RETUR
N
8801 IF cs=0 AND f1<5.5 THEN PR
INT AT 19,22; FLASH 1;" LASCIO!
": LET sg=sg+pt: GO TO 360
8802 IF f1=1 THEN PRINT AT 19,2
2; FLASH 1;" LASCIO! ": LET f1=0
: LET sg=sg+pt: GO TO 360
8803 IF f1=2 THEN LET pg=1000:
PRINT AT 18,22;"RILANCIO:";AT 19
,22;" $ ";pg;" ": LET pt=pt+pg:
GO TO 8830
8805 PRINT AT 18,22;"RILANCIO:"
8810 IF f1>=5 THEN LET hh=1: LE
T pg=(INT (RND*7)+4)*1000: PRINT
AT 19,22;" $ ";pg;" "; " AND p
q<10000: LET pt=pt+pg: GO TO 883
0
8820 IF f1=4 THEN LET pg=(INT (
RND*2)+1)*1000+(1000 AND cr>12 A
ND cs>=1)+(2000 AND cs=4): PRINT
AT 19,22;" $ ";pg;" ": LET pt=
pt+pg: GO TO 8830
8830 PRINT #0; BRIGHT 1; INVERSE
1;"A--->accetto","L--->lascio",
: PAUSE 0
8834 IF INKEY$<>"a" AND INKEY$<>
"1" THEN GO TO 8840
8840 PRINT AT 18,22; PAPER 0;"
":AT 19,22;"
8850 IF INKEY$="1" THEN PRINT A
T 2,22; FLASH 1;" LASCIO! ": INP
UT #0: GO TO 370
8860 LET sg=sg-pg: LET pt=pt+pg:
PRINT AT 11,25;pt
8870 PRINT AT 2,22; FLASH 1;" CI
STO! ": INPUT #0: RETURN
9000 FOR t=1 TO 4
9005 READ e$
9010 FOR n=0 TO 7
9020 READ e: POKE USR e$+n,e
9030 NEXT n: NEXT t
9040 RETURN
9100 DATA "a",0,108,254,254,124,
56,16,0
9110 DATA "b",0,16,56,124,254,12
4,56,16
9120 DATA "c",0,16,56,16,84,254,
84,16
9130 DATA "d",0,16,56,124,254,25
4,124,16
9500 REM SUBR per la preparazion
e della lavagna
9510 FOR b=0 TO 21

```

```

9520 PRINT AT b,21; PAPER 0;"
": NEXT b
9530 RETURN
9990 DATA CHR$ 144
9991 DATA CHR$ 145
9992 DATA CHR$ 146
9993 DATA CHR$ 147

```

Monoscopio

```

5 BRIGHT 1; PAPER 5; BORDER 7
: CLS : LET xc=127: LET yc=87: L
ET r=87
10 LET b=-2*yc
15 FOR x=xc-r TO xc+r
20 LET c=xc*xc+yc*yc-r*r-2*xc*
x+x**x
25 LET d=b*b-4*c
30 LET y1=(-b-SQR (d))/2
35 LET y2=INT (-b+SQR (d))/2
40 PLOT x,y1: DRAW 0,y2-y1
55 NEXT x
60 CIRCLE 127,87,87
110 DIM c(6): FOR k=1 TO 6: LET
c(k)=7-k: NEXT k
122 FOR m=0 TO 5: LET v=22723:
LET v=v+m*4
125 FOR k=0 TO 3
127 FOR j=1 TO 4
130 POKE v+j+32*k,BIN 01101000+
c(m+1)
135 NEXT j: NEXT k: NEXT m
150 DIM c(6): FOR k=1 TO 6: LET
c(k)=k: NEXT k
152 FOR m=0 TO 5: LET v=23043:
LET v=v+m*4
155 FOR k=0 TO 1
157 FOR j=1 TO 4
160 POKE v+j+32*k,BIN 01101000+
c(m+1)
165 NEXT j: NEXT k: NEXT m
170 FOR k=205 TO 176 STEP -3
175 PLOT PAPER 7; INVERSE 1;k,
48: DRAW PAPER 7; INVERSE 1;0,3
0
180 NEXT k
190 FOR k=175 TO 144 STEP -2
195 PLOT PAPER 7; INVERSE 1;k,
48: DRAW PAPER 7; INVERSE 1;0,3
0
200 NEXT k
210 FOR k=143 TO 112 STEP -1
212 IF k/3=INT (k/3) THEN GO T
O 220
215 PLOT INVERSE 1; PAPER 7;k,
48: DRAW PAPER 7; INVERSE 1;0,3
0
220 NEXT k
230 FOR k=108 TO 81 STEP -4
235 PLOT PAPER 7; INVERSE 1;k,

```

```

48: DRAW PAPER 7; INVERSE 1;0,3
0
236 PLOT PAPER 7; INVERSE 1;k+
1,48: DRAW PAPER 7; INVERSE 1;0
,30
240 NEXT k
250 FOR k=76 TO 48 STEP -8
255 FOR j=0 TO 3
260 PLOT INVERSE 1; PAPER 7;k+
j,48: DRAW PAPER 7; INVERSE 1;0
,30
265 NEXT j: NEXT k
268 LET o$=CHR$ 143+CHR$ 143: P
RINT AT 8,15;o$:AT 9,15;o$:AT 12
,15;o$:AT 13,15;o$
270 PLOT INVERSE 1; PAPER 7;40
,87: DRAW PAPER 7; INVERSE 1;17
4,0
275 FOR k=56 TO 208 STEP 16
285 PLOT PAPER 7; INVERSE 1;k,
79: DRAW PAPER 7; INVERSE 1;0,1
6
290 NEXT k
295 PLOT INVERSE 1; PAPER 7;12
8,55: DRAW INVERSE 1; PAPER 7;0
,56
300 FOR j=0 TO 6: FOR k=0 TO 9
310 PLOT INVERSE 1;63+k+j*20,1
28: DRAW INVERSE 1;0,11
320 NEXT k: NEXT j
322 LET v=22694: POKE v,BIN 011
01000: POKE v+1,BIN 01101000: LE
T v=22663: POKE v,BIN 01101000
330 FOR k=0 TO 8
340 PLOT INVERSE 1;88+k,140: D
RAW INVERSE 1;0,11
350 NEXT k
370 FOR k=0 TO 77
375 PLOT INVERSE 1;99+k,140: D
RAW INVERSE 1;0,11
380 NEXT k
500 LET v=22534: FOR k=0 TO 19:
FOR j=0 TO 2: POKE v+k+32*j,BIN
01101111: NEXT j: NEXT k
505 LET v=22633: FOR k=0 TO 13:
FOR j=0 TO 1: POKE v+k+32*j,BIN
01111000: NEXT j: NEXT k
507 LET v=22664: FOR k=0 TO 15:
FOR j=0 TO 1: POKE v+k+32*j,BIN
01111000: NEXT j: NEXT k
520 LET v=23110: FOR k=0 TO 19:
FOR j=0 TO 3: POKE v+k+32*j,BIN
01101100: NEXT j: NEXT k
550 PRINT AT 18,10;o$+o$+o$+o$+
o$+o$: AT 19,10;o$+o$+o$+o$+o$+o$
570 PRINT AT 1,12;o$+o$+o$+o$: A
T 2,12;o$+o$+o$+o$
580 PRINT INK 7; PAPER 0;AT 18
,12;"SINCLAIR";AT 19,12;"COMPUTE
R"
600 LET v=23183: FOR k=0 TO 1:
FOR j=0 TO 1: POKE v+k+32*j,BIN
01101010: NEXT j: NEXT k

```

```

610 LET v=23111: FOR k=0 TO 2:
FOR j=0 TO 1: POKE v+k+32*j,BIN
01101111: NEXT j: NEXT k
620 LET v=23126: FOR k=0 TO 2:
FOR j=0 TO 1: POKE v+k+32*j,BIN
01101111: NEXT j: NEXT k
700 STOP
1000 SAVE "monoscopio": STOP

```

Window

10 REM WINDOW

di SIMONE ZENDRINI

```

100 CLEAR 57699
110 LET a=0
120 FOR f=1 TO 221: READ b: LET
a=a+b: NEXT f
130 IF a<>24984 THEN CLS: PRI
NT AT 10,4;" HAI COMMESO UN ERR
ORE": STOP
140 PRINT AT 10,10;"TUTTO O.K"
150 RESTORE
260 DATA 58,148,227,167,254,8,5
6,11,254,16,56,12,214,16,17,0,80
,24,10,17,0,64
270 DATA 24,5,17,0,72,214,8,213
,22,0,95,33,0,0,6,32,25,16,253,2
09,25,58
280 DATA 149,227,22,0,95,25,229
,209,213,33,174,230,34,155,227,5
8,148,227,50,150,227,6
290 DATA 10,237,83,153,227,197,
6,10,197,237,83,151,227,6,8,205,
64,226,20,35,34,155
300 DATA 227,16,246,237,91,151,
227,19,193,16,231,237,91,153,227
,58,148,227,254,7,40,10
310 DATA 254,15,40,6,33,32,0,25
,24,4,33,32,7,25,229,209,33,148,
227,52,193,16
320 DATA 190,58,150,227,50,148,
227,225,0,0,0,58,182,225,50,27,2
26,124,230,24,203,47,203,47,203,
47,62,88
330 DATA 103,229,209,33,170,227
,34,155,227,58,166,225,71,197,23
7,83,151,227,58,173,225,71
340 DATA 205,108,226,19,42,155,
227,35,34,155,227,16,243,237,91,
151,227,33,32,0,25,229
350 DATA 209,193,16,221,201
360 DATA 26,42,155,227,119,201
370 DATA 42,155,227,126,18,201
380 FOR f=57700 TO 57908: READ
a: POKE f,a: NEXT f
390 FOR f=57920 TO 57925: READ
a: POKE f,a: NEXT f
400 FOR f=57930 TO 57935: READ

```



ilistati

```

a: POKE f,a: NEXT f
500 SAVE "WINDOW"CODE 57700,24
0
505 CLS
510 PRINT AT 10,10:"VERIFICA":
VERIFY "WINDOW"CODE 57700,240
520 CLS : PRINT AT 10,10:"TUTTO
O.K.": PAUSE 100: NEW
    
```

```

E164 3A94E3 LD A,(#E394)
E167 A7 AND A
E168 FE00 CP #00
E16A 3800 JR C,LE177
E16C FE10 CP #10
E16E 380C JR C,LE17C
E170 D610 SUB #10
E172 110050 LD DE,#5000
E175 180A JR LE181
E177 110040 LE177 LD DE,#4000
E17A 1805 JR LE181
E17C 110048 LE17C LD DE,#4800
E17F D608 SUB #08
E181 D5 LE181 PUSH DE
E182 1600 LD D,#00
E184 5F LD E,A
E185 210000 LD HL,#0000
E188 0620 LD B,#20
E18A 19 LE18A ADD HL,DE
E18B 10FD DJNZ LE18A
E18D D1 POP DE
E18E 19 ADD HL,DE
E18F 3A95E3 LD A,(#E395)
E192 1600 LD D,#00
E194 5F LD E,A
E195 19 ADD HL,DE
E196 E5 PUSH HL
E197 D1 POP DE
E198 D5 PUSH DE
E199 21AE6 LD HL,#E6AE
E19C 2298E3 LD (#E398),HL
E19F 3A94E3 LD A,(#E394)
E1A2 3296E3 LD (#E396),A
E1A5 060A LD B,#0A
E1A7 ED5399E3 LE1A7 LD (#E399),DE
E1AB C5 PUSH BC
E1AC 060A LD B,#0A
E1AE C5 LE1AE PUSH BC
E1AF ED5397E3 LD (#E397),DE
E1B3 0608 LD B,#08
E1B5 CD40E2 LE1B5 CALL LE240
E1B8 14 INC D
E1B9 23 INC HL
E1BA 2298E3 LD (#E398),HL
E1BD 10F6 DJNZ LE1B5
E1BF ED5B97E3 LD DE,(#E397)
E1C3 13 INC DE
E1C4 C1 POP BC
E1C5 10E7 DJNZ LE1AE
E1C7 ED5B99E3 LD DE,(#E399)
E1CB 3A94E3 LD A,(#E394)
E1CE FE07 CP #07
E1D0 280A JR Z,LE1DC
E1D2 FE0F CP #0F
E1D4 2806 JR Z,LE1DC
E1D6 212000 LD HL,#0020
E1D9 19 ADD HL,DE
E1DA 1804 JR LE1E0
    
```

```

E1DC 212007 LE1DC LD HL,#0720
E1DF 19 ADD HL,DE
E1E0 E5 PUSH HL
E1E1 D1 POP DE
E1E2 2194E3 LD HL,#E394
E1E5 34 INC (HL)
E1E6 C1 POP BC
E1E7 10BE DJNZ LE1A7
E1E9 3A96E3 LD A,(#E396)
E1EC 3294E3 LD (#E394),A
E1EF E1 POP HL
E1F0 00 NOP
E1F1 00 NOP
E1F2 00 NOP
E1F3 3AB6E1 LD A,(LE1B6)
E1F6 3218E2 LD (LE21B),A
E1F9 7C LD A,H
E1FA E618 AND #18
E1FC CB2F SRA A
E1FE CB2F SRA A
E200 CB2F SRA A
E202 3E58 LD A,#58
E204 67 LD H,A
E205 E5 PUSH HL
E206 D1 POP DE
E207 21AAE3 LD HL,#E3AA
E20A 2298E3 LD (#E398),HL
E20D 3AA6E1 LD A,(LE1A6)
E210 47 LD B,A
E211 C5 LE211 PUSH BC
E212 ED5397E3 LD HL,#E397,DE
E216 3AADE1 LD A,(LE1AD)
E219 47 LD B,A
E21A CD6CE2 LE21A CALL #E26C
E21D 13 INC DE
E21E 2A98E3 LD HL,(#E398)
E221 23 INC HL
E222 2298E3 LD HL,#E398,HL
E225 10F3 DJNZ LE21A
E227 ED5B97E3 LD DE,(#E397)
E228 212000 LD HL,#0020
E22E 19 ADD HL,DE
E22F E5 PUSH HL
E230 D1 POP DE
E231 C1 POP BC
E232 10DD DJNZ LE211
E234 C9 RET
E235 00 NOP
E236 00 NOP
E237 00 NOP
E238 00 NOP
E239 00 NOP
E23A 00 NOP
E23B 00 NOP
E23C 00 NOP
E23D 00 NOP
E23E 00 NOP
E23F 00 NOP
E240 1A LE240 LD A,(DE)
E241 2A98E3 LD HL,(#E398)
E244 77 LD (HL),A
E245 C9 RET
E246 00 NOP
E247 00 NOP
E248 00 NOP
E249 00 NOP
E24A 2A98E3 LD HL,(#E398)
E24D 7E LD A,(HL)
E24E 12 LD (DE),A
E24F C9 RET
    
```

Il movimento sullo schermo

10 REM MOVCHload: caricamento
c.m. a indir."n",con rilocazione

11 REM variare CLEAR e LET n
di linea 20 secondo fabbisogno

20 CLEAR 32164: LET n=32165

30 FOR x=0 TO 434: READ y: POK
E n+x,y: NEXT x

40 POKE n+34,INT ((n+105)/256)
: POKE n+33,n+105-256*PEEK (n+34)

50 POKE n+37,INT ((n+403)/256)
: POKE n+36,n+403-256*PEEK (n+37)

60 POKE n+52,INT ((n+252)/256)
: POKE n+51,n+252-256*PEEK (n+52)

70 POKE n+68,INT ((n+136)/256)
: POKE n+67,n+136-256*PEEK (n+68)

80 POKE n+93,INT ((n+284)/256)
: POKE n+92,n+284-256*PEEK (n+93)

90 POKE n+187,INT ((n+125)/256)
: POKE n+186,n+125-256*PEEK (n+187)

100 POKE n+48,PEEK (n+33): POKE
n+49,PEEK (n+34): POKE n+64,PEEK
K (n+33): POKE n+65,PEEK (n+34):
POKE n+89,PEEK (n+33): POKE n+9
0,PEEK (n+34)

110 POKE n+269,PEEK (n+186): PO
KE n+270,PEEK (n+187): POKE n+33
4,PEEK (n+186): POKE n+335,PEEK
(n+187): POKE n+420,PEEK (n+186)
: POKE n+421,PEEK (n+187)

120 PRINT "SAVE ""MOVCHAR"" CO
DE ";n;" ,435"

130 SAVE "MOVCHAR"CODE n,435

401 DATA 1,254,239,237,120,6,24
7,237,96,31,1,231,3,208,203,100,
42,0,91,31,31,48,17,31,48,29,31,
48,51,32,225

402 DATA 37,205,14,126,205,56,1
27,24,216

403 DATA 36,31,48,27,31,48,49

404 DATA 205,14,126,205,161,126
,24,201

405 DATA 40,18

406 DATA 62,224,50,129,92,45,20
5,14,126,205,45,126,24,185

407 DATA 62,225,24,240

408 DATA 62,223,37,24,235

409 DATA 40,18

410 DATA 62,32,50,129,92,44,205
,14,126,205,193,126,24,160

411 DATA 62,33,24,240

412 DATA 62,31,37,24,235

500 DATA 76,6,88,38,0,41,41,
41,41,9,126,0,167,192,193,6,0,78
,201

550 DATA 14,3,6,0,16,254,65,13,
16,248,201

600 DATA 33,0,0,1,0,8,58,0,91,1
44,40,3,144,32,2

601 DATA 203,193,197,22,6,203,6
5,40,2,22,255

602 DATA 58,129,92,95,24,3

603 DATA 84,93,21

604 DATA 229,213,237,91,176,92,
25,126,225,25,119,225,16,47,205,
34,126,229,25,54,0,225,193,22,25
5,203,65,40,2,22,248

605 DATA 58,129,92,95,25,16,18,
237,91,176,92,25,34,176,92,33,0,
91,53,214,224,35,134,119,201

606 DATA 197,6,8,24,192

607 DATA 124,214,6,40,7,214,249
,40,6

608 DATA 36,24,180

609 DATA 181,40,250

610 DATA 33,0,0,24,158

700 DATA 62,8

701 DATA 42,176,92,6,8

702 DATA 203,62,44,203,30,36,45
16,247,205,34,126,71,61,16,235,
33,176,92,52,33,1,91,52,201

800 DATA 33,0,7,1,0,8,58,0,91,1
48,40,3,144,32,2

801 DATA 203,193,197,22,0,203,6
5,40,2,22,7

802 DATA 58,129,92,95,24,3

803 DATA 84,93,20

804 DATA 229,213,237,91,176,92,
25,126,225,25,119,225,16,51,205,
34,126,229,25,54,0,225,193,22,0,
203,65,40,2,22,7

805 DATA 58,129,92,95,25,16,22,
237,91,176,92,25,17,0,249,25,34,
176,92,33,0,91,52,214,32,35,134,
119,201

806 DATA 197,6,8,24,188

807 DATA 124,167,40,10,214,7,40
3

808 DATA 37,24,177

809 DATA 181,40,250

810 DATA 33,0,7,24,155

900 DATA 62,8

901 DATA 42,176,92,6,8

902 DATA 203,38,45,203,22,36,44
16,247,205,34,126,71,61,16,235,
33,176,92,53,33,1,91,53,201

927E 01FEFF	10 KTEST	LD	BC,61438
9281 ED78	11	IN	A,(C)
9283 06F7	12	LD	B,247



ilistati

9285 ED60	13	IN	H, (C)	92EC 29	66	ADD	HL, HL
9287 1F	14	RRA		92ED 29	67	ADD	HL, HL
9288 01E703	15	LD	BC, 999	92EE 29	68	ADD	HL, HL
928B D0	16	RET	NC	92EF 29	69	ADD	HL, HL
928C CB64	17	BIT	4, H	92F0 29	70	ADD	HL, HL
928E 2A005B	18	LD	HL, (23296)	92F1 09	71	ADD	HL, BC
9291 1F	19	RRA		92F2 7E	72	LD	A, (HL)
9292 1F	20	RRA		92F3 00	73	NOP	
9293 3011	21	JR	NC, K8	92F4 A7	74	AND	A
9295 1F	22	RRA		92F5 C0	75	RET	NZ
9296 301D	23	JR	NC, K7	92F6 C1	76	POP	BC
9298 1F	24	RRA		92F7 0600	77	LD	B, 0
9299 3033	25	JR	NC, K6	92F9 4E	78	LD	C, (HL)
929B 20E1	26	JR	NZ, KTEST	92FA C9	79	RET	
929D 25	27	OV	DEC	92FB 0E03	80	WAIT	LD C, 3
929E CDE792	28	CALL	MOVST	92FD 0600	81	WAA	LD B, 0
92A1 CD1194	29	CALL	MOV0V	92FF 10FE	82	WBB	DJNZ WBB
92A4 18DB	30	JR	KTEST	9301 41	83	LD	B, C
92A6 24	31	K8	INC	9302 0D	84	DEC	C
92A7 1F	32	RRA		9303 10F8	85	DJNZ	WAA
92AB 301B	33	JR	NC, NEST	9305 C9	86	RET	
92AA 1F	34	RRA		9306 210000	87	MVNORD	LD HL, 0
92AB 3031	35	JR	NC, SEST	9309 010008	88	LD	BC, 2048
92AD CDE792	36	EST	CALL	930C 3A005B	89	LD	A, (23296)
92B0 CD7A93	37	CALL	MOVEST	930F 90	90	SUB	B
92B3 18C9	38	JR	KTEST	9310 2803	91	JR	Z, NAA
92B5 2812	39	K7	JR	9312 90	92	SUB	B
92B7 3EE0	40	NORD	LD	9313 2802	93	JR	Z, NBB
92B9 32815C	41	NX	LD	9315 CBC1	94	NAA	SET 0, C
92BC 2D	42	DEC	L	9317 C5	95	NBB	PUSH BC
92BD CDE792	43	CALL	MOVST	9318 1606	96	NCC	LD D, 6
92C0 CD0693	44	CALL	MVNORD	931A CB41	97	BIT	0, C
92C3 18B9	45	JR	KTEST	931C 2802	98	JR	Z, NDD
92C5 3EE1	46	NEST	LD	931E 16FF	99	LD	D, 255
92C7 18F0	47	JR	NX	9320 3A815C	100	NDD	LD A, (23681)
92C9 3EDF	48	NOV	LD	9323 5F	101	LD	E, A
92CB 25	49	DEC	H	9324 1803	102	JR	NFF
92CC 18EB	50	JR	NX	9326 54	103	NEE	LD D, H
92CE 2812	51	K6	JR	9327 5D	104	LD	E, L
92D0 3E20	52	SUD	LD	9328 15	105	DEC	D
92D2 32815C	53	SX	LD	9329 E5	106	NFF	PUSH HL
92D5 2C	54	INC	L	932A D5	107	PUSH	DE
92D6 CDE792	55	CALL	MOVST	932B ED5B005C	108	LD	DE, (23728)
92D9 CD9A93	56	CALL	MOV0SUD	932F 19	109	ADD	HL, DE
92DC 18A0	57	JR	KTEST	9330 7E	110	LD	A, (HL)
92DE 3E21	58	SEST	LD	9331 E1	111	POP	HL
92E0 18F0	59	JR	SX	9332 19	112	ADD	HL, DE
92E2 3E1F	60	SOV	LD	9333 77	113	LD	(HL), A
92E4 25	61	DEC	H	9334 E1	114	POP	HL
92E5 18EB	62	JR	SX	9335 102F	115	DJNZ	NLL
92E7 4C	63	MOVST	LD	9337 CDFB92	116	CALL	WAIT
92E8 0658	64	LD	B, 88	933A E5	117	PUSH	HL
92EA 2600	65	LD	H, 0	933B 19	118	ADD	HL, DE

i listati

933C 3600	119	LD (HL),0	939A 210007	172 MOVSD	LD HL,1792
933E E1	120	POP HL	939D 010008	173	LD BC,2048
933F C1	121	POP BC	93A0 3A005B	174	LD A,(23296)
9340 16FF	122	LD D,255	93A3 94	175	SUB H
9342 CB41	123	BIT 0,C	93A4 2803	176	JR Z,SA
9344 2002	124	JR Z,NGG	93A6 90	177	SUB B
9346 16F8	125	LD D,248	93A7 2002	178	JR NZ,SBB
9348 3A815C	126 NGG	LD A,(23681)	93A9 CBC1	179 SAA	SET 0,C
9348 5F	127	LD E,A	93AB C5	180 SBB	PUSH BC
934C 19	128	ADD HL,DE	93AC 1600	181 SCC	LD D,0
934D 1012	129	DJNZ NHH	93AE CB41	182	BIT 0,C
934F ED5BB05C	130	LD DE,(23728)	93B0 2802	183	JR Z,SD
9353 19	131	ADD HL,DE	93B2 1607	184	LD D,7
9354 22B05C	132	LD (23728),HL	93B4 3A815C	185 SDD	LD A,(23681)
9357 21005B	133	LD HL,23296.	93B7 5F	186	LD E,A
935A 35	134	DEC (HL)	93B8 1803	187	JR SFF
935B D6E0	135	SUB 224	93BA 54	188 SEE	LD D,H
935D 23	136	INC HL	93BB 5D	189	LD E,L
935E 86	137	ADD A,(HL)	93BC 14	190	INC D
935F 77	138	LD (HL),A	93BD E5	191 SFF	PUSH HL
9360 C9	139	RET	93BE 05	192	PUSH DE
9361 C5	140 NHH	PUSH BC	93BF ED5BB05C	193	LD DE,(23728)
9362 0600	141	LD B,B	93C3 19	194	ADD HL,DE
9364 18C0	142	JR NEE	93C4 7E	195	LD A,(HL)
9366 7C	143 NLL	LD A,H	93C5 E1	196	POP HL
9367 D606	144	SUB 6	93C6 19	197	ADD HL,DE
9369 2807	145	JR Z,NNN	93C7 77	198	LD (HL),A
936B D6F9	146	SUB 249	93C8 E1	199	POP HL
936D 2806	147	JR Z,NPP	93C9 1033	200	DJNZ SLL
936F 24	148 NMM	INC H	93CB CDFB92	201	CALL WAIT
9370 18B4	149	JR NEE	93CE E5	202	PUSH HL
9372 B5	150 NNN	OR L	93CF 19	203	ADD HL,DE
9373 28FA	151	JR Z,NMM	93D0 3600	204	LD (HL),0
9375 210000	152 NPP	LD HL,0	93D2 E1	205	POP HL
9378 189E	153	JR NCC	93D3 C1	206	POP BC
937A 3E08	154 MOVEST	LD A,B	93D4 1600	207	LD D,0
937C 2AB05C	155 EAA	LD HL,(23728)	93D6 CB41	208	BIT 0,C
937F 0600	156	LD B,B	93D8 2802	209	JR Z,SGG
9381 CB3E	157 EBB	SRL (HL)	93DA 1607	210	LD D,7
9383 2C	158	INC L	93DC 3A815C	211 SGG	LD A,(23681)
9384 CB1E	159	RR (HL)	93DF 5F	212	LD E,A
9386 24	160	INC H	93E0 19	213	ADD HL,DE
9387 2D	161	DEC L	93E1 1016	214	DJNZ SHH
9388 10F7	162	DJNZ EBB	93E3 ED5BB05C	215	LD DE,(23728)
938A CDFB92	163	CALL WAIT	93E7 19	216	ADD HL,DE
938D 47	164	LD B,A	93E8 1100F9	217	LD DE,-1792
938E 3D	165	DEC A	93EB 19	218	ADD HL,DE
938F 10EB	166	DJNZ EAA	93EC 22B05C	219	LD (23728),HL
9391 21B05C	167	LD HL,23728	93EF 21005B	220	LD HL,23296
9394 34	168	INC (HL)	93F2 34	221	INC (HL)
9395 21015B	169	LD HL,23297	93F3 D620	222	SUB 32
9398 34	170	INC (HL)	93F5 23	223	INC HL
9399 C9	171	RET	93F6 86	224	ADD A,(HL)



```

93F7 77      225      LD (HL),A
93F8 C9      226      RET
93F9 C5      227 SHH   PUSH BC
93FA 0608    228      LD B,B
93FC 18BC    229      JR SEE
93FE 7C      230 SLL   LD A,H
93FF A7      231      AND A
9400 200A    232      JR Z,SPP
9402 D607    233      SUB 7
9404 2003    234      JR Z,SNN
9406 25      235 SMM   DEC H
9407 18B1    236      JR SEE
9409 85      237 SNN   OR L
940A 20FA    238      JR Z,SMM
940C 210007  239 SPP   LD HL,1792
940F 189B    240      JR SCC
9411 3E08    241 MOV DV LD A,B
9413 2A05C   242 OAA   LD HL,(23728)
9416 0608    243      LD B,B
9418 CB26    244 OBB   SLA (HL)
941A 2D      245      DEC L
941B CB16    246      RL (HL)
941D 24      247      INC H
941E 2C      248      INC L
941F 10F7    249      DJNZ OBB
9421 CDFB92  250      CALL WAIT
9424 47      251      LD B,A
9425 3D      252      DEC A
9426 10EB    253      DJNZ OAA
9428 21B05C  254      LD HL,23728
942B 35      255      DEC (HL)
942C 21015B  256      LD HL,23297
942F 35      257      DEC (HL)
9430 C9      258      RET
    
```

```

1 REM MOVCHAR test program
10 CLEAR 32164
20 PRINT AT 2,2:"loading ""MOV
CHAR""": LOAD "MOVCHAR"CODE
21 PRINT ; FLASH 1;"STOP THE
TAPE": PAUSE 100: CLS
30 REM print sbarramenti con
attributo 0
31 FOR n=0 TO 31: POKE 22560+n
,0: POKE 23263-n,0: NEXT n
32 FOR n=0 TO 608 STEP 32: POK
E 23200-n,0: POKE 22623+n,0: NEX
T n
33 FOR n=0 TO 11: POKE 22746-n
,0: POKE 23109+n,0: NEXT n
34 FOR n=0 TO 224 STEP 32: POK
E 22728+n,0: POKE 23127-n,0: NEX
T n
40 REM print start position
41 LET riga=12: LET col=16
42 LET ind=16384+col+(riga-INT
    
```

```

(riga/B)*8)*32+INT (riga/B)*204
B 43 POKE 23728,ind-INT (ind/256
)*256: POKE 23729,INT (ind/256)
44 POKE 23296,riga: POKE 23297
,col
45 PRINT AT riga,col;"*"
46 LET count=0
50 REM chiamata al programma
codice macchina
51 LET RC=USR 32165
60 REM test codice di ritorno
61 IF RC=999 THEN GO TO 80
70 REM gestione ritorno da
"scontro" (RC=0)
71 LET n=22528+32*PEEK 23296+P
EEK 23297
72 POKE n,PEEK n+128
73 LET count=count+1
74 PRINT #1;AT 1,0;"SCONTR0 !"
,count;" penalita'"
75 PAUSE 30: PAUSE 0: POKE n,P
EEK n-128
76 PRINT #1;AT 1,0;"
": GO TO 50
80 REM gestione ritorno da
"fuoco!" (RC=999)
81 PRINT AT PEEK 23296,PEEK 23
297; INVERSE 1;" "
82 PRINT AT 0,0;"FUOCO !"
83 PAUSE 30: PAUSE 0: CLS : GO
TO 30
    
```

Doppia copia

```

FA00          10      ORG 64000
FA00 D9      20      INIT   EXX
FA01 E5      30      PUSH HL
FA02 D9      40      EXX
FA03 3E1B    50      LD A,27
FA05 CD28FA  60      CALL FUOR
FA08 3E4D    70      LD A,77
FA0A CD28FA  80      CALL FUOR
FA0D 3E01    90      LD A,1
FA0F CD28FA 100      CALL FUOR
FA12 3E1B   110      LD A,27
FA14 CD28FA 120      CALL FUOR
FA17 3E41   130      LD A,65
FA19 CD28FA 140      CALL FUOR
FA1C 3E08   150      LD A,B
FA1E CD28FA 160      CALL FUOR
FA21 CD84FA 170      CALL TERZ
FA24 D9     180      EXX
FA25 E1     190      POP HL
FA26 D9     200      EXX
FA27 C9     210      RET
FA28 D9     220      FUOR   EXX
FA29 F5     230      PUSH AF
FA2A CF     240      RST 8
FA2B 1E     250      DEFB #1E
FA2C D9     260      EXX
FA2D F1     270      POP AF
    
```

listabi

FA2E C9	280	RET	FA9C 0608	920 UNTE	LD B,8
FA2F 14	290 SHF1	INC D	FA9E CD75FA	930 EXT1	CALL LINH
FA30 14	300	INC D	FAA1 C5	940	PUSH BC
FA31 14	310	INC D	FAA2 0620	950	LD B,32
FA32 14	320	INC D	FAA4 1B	960 MEM1	DEC DE
FA33 C9	330	RET	FAA5 10FD	970	DJNZ MEM1
FA34 15	340 SHF2	DEC D	FAA7 C1	980	POP BC
FA35 15	350	DEC D	FAA8 CD2FFA	990	CALL SHF1
FA36 15	360	DEC D	FAAB CD75FA	1000	CALL LINH
FA37 15	370	DEC D	FAAE CD34FA	1010	CALL SHF2
FA38 C9	380	RET	FAB1 10EB	1020	DJNZ EXT1
FA39 C9	390 SOMM	INC A	FAB3 C9	1030	RET
FA3A C9	400	RET	FAB4 110040	1040 TERZ	LD DE,16384
FA3B 210000	410 CICL	LD HL,0	FAB7 CD9CFA	1050	CALL UNTE
FA3E 19	420	ADD HL,DE	FABA 110048	1060	LD DE,18432
FA3F C5	430	PUSH BC	FABD CD9CFA	1070	CALL UNTE
FA40 0604	440	LD B,4	FAC0 110050	1080	LD DE,20480
FA42 AF	450	XOR A	FAC3 CD9CFA	1090	CALL UNTE
FA43 C5	460 NEXT	PUSH BC	FAC6 C9	1100	RET
FA44 0602	470	LD B,2	FAC7	1110	END
FA46 B7	480 INT7	ADD A,A			
FA47 CB7E	490 CHG1	BIT 7,(HL)			
FA49 C439FA	500	CALL NZ,SOMM	7E2C	10	ORG 32300
FA4C 10F8	510	DJNZ INT7	7E2C D9	20 INIT	EXX
FA4E C1	520	POP BC	7E2D E5	30	PUSH HL
FA4F 24	530	INC H	7E2E D9	40	EXX
FA50 10F1	540	DJNZ NEXT	7E2F 3E1B	50	LD A,27
FA52 CD28FA	550	CALL FUOR	7E31 CD547E	60	CALL FUOR
FA55 CD28FA	560	CALL FUOR	7E34 3E4D	70	LD A,77
FA5B C1	570	POP BC	7E36 CD547E	80	CALL FUOR
FA59 C9	580	RET	7E39 3E01	90	LD A,1
FA5A C5	590 CARH	PUSH BC	7E3B CD547E	100	CALL FUOR
FA5B 0608	600	LD B,8	7E3E 3E1B	110	LD A,27
FA5D CD3BFA	610 GIRA	CALL CICL	7E40 CD547E	120	CALL FUOR
FA60 C5	620	PUSH BC	7E43 3E41	130	LD A,65
FA61 0608	630	LD B,8	7E45 CD547E	140	CALL FUOR
FA63 3A48FA	640	LD A,(CHG1+1)	7E48 3E08	150	LD A,8
FA66 90	650	SUB B	7E4A CD547E	160	CALL FUOR
FA67 3248FA	660	LD (CHG1+1),A	7E4D CDE07E	170	CALL TERZ
FA6A C1	670	POP BC	7E50 D9	180	EXX
FA6B 10F0	680	DJNZ GIRA	7E51 E1	190	POP HL
FA6D 3E7E	690	LD A,126	7E52 D9	200	EXX
FA6F 3248FA	700	LD (CHG1+1),A	7E53 C9	210	RET
FA72 C1	710	POP BC	7E54 D9	220 FUOR	EXX
FA73 13	720	INC DE	7E55 F5	230	PUSH AF
FA74 C9	730	RET	7E56 CF	240	RST 8
FA75 C5	740 LINH	PUSH BC	7E57 1E	250	DEFB #1E
FA76 3E1B	750	LD A,27	7E58 D9	260	EXX
FA78 CD28FA	760	CALL FUOR	7E59 F1	270	POP AF
FA7B 3E48	770	LD A,75	7E5A C9	280	RET
FA7D CD28FA	780	CALL FUOR	7E5B 14	290 SHF1	INC D
FA80 AF	790	XOR A	7E5C 14	300	INC D
FAB1 CD28FA	800	CALL FUOR	7E5D 14	310	INC D
FAB4 3E02	810	LD A,2	7E5E 14	320	INC D
FAB6 CD28FA	820	CALL FUOR	7E5F C9	330	RET
FAB9 0620	830	LD B,32	7E60 15	340 SHF2	DEC D
FABB CD5AFA	840 RIGH	CALL CARH	7E61 15	350	DEC D
FABE 10FB	850	DJNZ RIGH	7E62 15	360	DEC D
FA90 3E0D	860	LD A,13	7E63 15	370	DEC D
FA92 CD28FA	870	CALL FUOR	7E64 C9	380	RET
FA95 3E0A	880	LD A,10	7E65 3C	390 SOMM	INC A
FA97 CD28FA	890	CALL FUOR	7E66 C9	400	RET
FA9A C1	900	POP BC	7E67 210000	410 CICL	LD HL,0
FA9B C9	910	RET	7E6A 19	420	ADD HL,DE



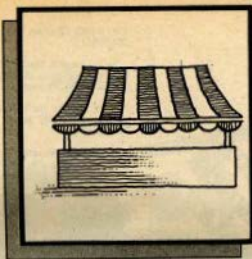
LISTA

7E6B C5	430	PUSH BC	7EE9 CDC87E	1070	CALL UNTE
7E6C 0604	440	LD B,4	7EEC 110050	1080	LD DE,20480
7E6E AF	450	XOR A	7EEF CDC87E	1090	CALL UNTE
7E6F C5	460	NEXT PUSH BC	7EF2 C9	1100	RET
7E70 0602	470	LD B,2	7EF3	1110	END
7E72 87	480	ADD A,A			
7E73 CB7E	490	CHG1 BIT 7,(HL)			
7E75 C4657E	500	CALL NZ,SOMM			
7E78 10FB	510	DJNZ INT7			
7E7A C1	520	POP BC			
7E7B 24	530	INC H			
7E7C 10F1	540	DJNZ NEXT			
7E7E CD547E	550	CALL FUOR			
7E81 CD547E	560	CALL FUOR			
7E84 C1	570	POP BC			
7E85 C9	580	RET			
7E86 C5	590	CARH PUSH BC			
7E87 0608	600	LD B,8			
7E89 CD677E	610	GIRA CALL C1CL			
7E8C C5	620	PUSH BC			
7E8D 0608	630	LD B,8			
7E8F 3A747E	640	LD A,(CHG1+1)			
7E92 90	650	SUB B			
7E93 32747E	660	LD (CHG1+1),A			
7E96 C1	670	POP BC			
7E97 10FB	680	DJNZ GIRA			
7E99 3E7E	690	LD A,126			
7E9B 32747E	700	LD (CHG1+1),A			
7E9E C1	710	POP BC			
7E9F 13	720	INC DE			
7EA0 C9	730	RET			
7EA1 C5	740	LINH PUSH BC			
7EA2 3E1B	750	LD A,27			
7EA4 CD547E	760	CALL FUOR			
7EA7 3E4B	770	LD A,75			
7EA9 CD547E	780	CALL FUOR			
7EAC AF	790	XOR A			
7EAD CD547E	800	CALL FUOR			
7EB0 3E02	810	LD A,2			
7EB2 CD547E	820	CALL FUOR			
7EB5 0620	830	LD B,32			
7EB7 CDB67E	840	RIGH CALL CARH			
7EBA 10FB	850	DJNZ RIGH			
7EBC 3E0D	860	LD A,13			
7EBE CD547E	870	CALL FUOR			
7EC1 3E0A	880	LD A,10			
7EC3 CD547E	890	CALL FUOR			
7EC6 C1	900	POP BC			
7EC7 C9	910	RET			
7EC8 0608	920	UNTE LD B,8			
7ECA CDA17E	930	EXT1 CALL LINH			
7ECD C5	940	PUSH BC			
7ECE 0620	950	LD B,32			
7ED0 1B	960	MEM1 DEC DE			
7ED1 10FD	970	DJNZ MEM1			
7ED3 C1	980	POP BC			
7ED4 CD597E	990	CALL SHF1			
7ED7 CDA17E	1000	CALL LINH			
7EDA CD607E	1010	CALL SHF2			
7EDD 10EB	1020	DJNZ EXT1			
7EDF C9	1030	RET			
7EE0 110040	1040	TERZ LD DE,16384			
7EE3 CDC87E	1050	CALL UNTE			
7EE6 11004B	1060	LD DE,10432			
			FACB	10	DRG 64200
			FACB C345FB	20	JP INIT
			FACB D9	30	EXI
			FACC CF	40	RST 8
			FACD 1E	50	DEFB #1E
			FACE D9	60	EXI
			FACF C9	70	RET
			FADD 3C	80	SOMM INC A
			FAD1 C9	90	RET
			FAD2 CB7E	100	SETT BIT 7,(HL)
			FAD4 C4D0FA	110	CALL NZ,SOMM
			FAD7 24	120	INC H
			FADD C9	130	RET
			FAD9 C5	140	CAR PUSH BC
			FADA 0608	150	LD B,8
			FADC C5	160	CAR1 PUSH BC
			FADD 210000	170	LD HL,0
			FAE0 19	180	ADD HL,DE
			FAE1 0607	190	LD B,7
			FAE3 AF	200	XOR A
			FAE4 CDD2FA	210	INT7 CALL SETT
			FAE7 87	220	ADD A,A
			FAE8 10FA	230	DJNZ INT7
			FAEA CDD2FA	240	CALL SETT
			FAED CDCBFA	250	CALL FUOR
			FAF0 3AD3FA	260	LD A,(SETT+1)
			FAF3 0608	270	LD B,8
			FAF5 90	280	SUB B
			FAF6 32D3FA	290	LD (SETT+1),A
			FAF9 C1	300	POP BC
			FAFA 10E0	310	DJNZ CAR1
			FAFC 3E7E	320	LD A,126
			FAFE 32D3FA	330	LD (SETT+1),A
			FB01 13	340	INC DE
			FB02 C1	350	POP BC
			FB03 C9	360	RET
			FB04 C5	370	LINE PUSH BC
			FB05 3E1B	380	LD A,27
			FB07 CDCBFA	390	CALL FUOR
			FB0A 3E4B	400	LD A,75
			FB0C CDCBFA	410	CALL FUOR
			FB0F AF	420	XOR A
			FB10 CDCBFA	430	CALL FUOR
			FB13 3C	440	INC A
			FB14 CDCBFA	450	CALL FUOR
			FB17 0620	460	LD B,32
			FB19 CDD9FA	470	NEWC CALL CAR
			FB1C 10FB	480	DJNZ NEWC
			FB1E 3E0D	490	LD A,13
			FB20 CDCBFA	500	CALL FUOR
			FB23 3E0A	510	LD A,10
			FB25 CDCBFA	520	CALL FUOR
			FB28 C1	530	POP BC
			FB29 C9	540	RET
			FB2A 0608	550	UNTE LD B,8
			FB2C CD04FB	560	EXT1 CALL LINE
			FB2F 10FB	570	DJNZ EXT1
			FB31 C9	580	RET
			FB32 110040	590	TERZ LD DE,16384

LISTA 1

FB35	CD2AFB	600		CALL	UNTE	7F2D 13	340	INC	DE
FB38	11004B	610		LD	DE,1B432	7F2E C1	350	POP	BC
FB38	CD2AFB	620		CALL	UNTE	7F2F C9	360	RET	
FB3E	110050	630		LD	DE,20400	7F30 C5	370	PUSH	BC
FB41	CD2AFB	640		CALL	UNTE	7F31 3E1B	380	LD	A,27
FB44	C9	650		RET		7F33 CDF77E	390	CALL	FUOR
FB45	F5	660	INIT	PUSH	AF	7F36 3E4B	400	LD	A,75
FB46	D9	670		EXX		7F38 CDF77E	410	CALL	FUOR
FB47	E5	680		PUSH	HL	7F3B AF	420	XOR	A
FB48	D9	690		EXX		7F3C CDF77E	430	CALL	FUOR
FB49	3E7E	700		LD	A,126	7F3F 3C	440	INC	A
FB48	32D3FA	710		LD	(SETT+1),A	7F40 CDF77E	450	CALL	FUOR
FB4E	3E1B	720		LD	A,27	7F43 0620	460	LD	B,32
FB50	CDCBFA	730		CALL	FUOR	7F45 CD057F	470	NEWC	CALL CAR
FB53	3E4D	740		LD	A,77	7F48 10FB	480	DJNZ	NEWC
FB55	CDCBFA	750		CALL	FUOR	7F4A 3E0D	490	LD	A,13
FB58	AF	760		XOR	A	7F4C CDF77E	500	CALL	FUOR
FB59	CDCBFA	770		CALL	FUOR	7F4F 3E0A	510	LD	A,10
FB5C	3E1B	780		LD	A,27	7F51 CDF77E	520	CALL	FUOR
FB5E	CDCBFA	790		CALL	FUOR	7F54 C1	530	POP	BC
FB61	3E41	800		LD	A,65	7F55 C9	540	RET	
FB63	CDCBFA	810		CALL	FUOR	7F56 0608	550	UNTE	LD B,8
FB66	3E0B	820		LD	A,8	7F58 CD307F	560	EXT1	CALL LINE
FB6B	CDCBFA	830		CALL	FUOR	7F5B 10FB	570	DJNZ	EXT1
FB6B	CD32FB	840		CALL	TERZ	7F5D C9	580	RET	
FB6E	D9	850		EXX		7F5E 110040	590	TERZ	LD DE,16304
FB6F	E1	860		POP	HL	7F61 CD567F	600	CALL	UNTE
FB70	D9	870		EXX		7F64 11004B	610	LD	DE,18432
FB71	F1	880		POP	AF	7F67 CD567F	620	CALL	UNTE
FB72	C9	890		RET		7F6A 110050	630	LD	DE,20400
FB73		900		END		7F6D CD567F	640	CALL	UNTE
						7F70 C9	650	RET	
7EF4		10		ORG	32500	7F71 F5	660	INIT	PUSH AF
7EF4	C3717F	20		JP	INIT	7F72 D9	670	EXX	
7EF7	D9	30	FUOR	EXX		7F73 E5	680	PUSH	HL
7EF8	CF	40		RST	B	7F74 D9	690	EXX	
7EF9	1E	50		DEFB	#1E	7F75 3E7E	700	LD	A,126
7EFA	D9	60		EXX		7F77 32FF7E	710	LD	(SETT+1),A
7EFB	C9	70		RET		7F7A 3E1B	720	LD	A,27
7EFC	3C	80	SOMM	INC	A	7F7C CDF77E	730	CALL	FUOR
7EFD	C9	90		RET		7F7F 3E4D	740	LD	A,77
7EFE	C87E	100	SETT	BIT	7,(HL)	7F81 CDF77E	750	CALL	FUOR
7F00	C4FC7E	110		CALL	NZ,SOMM	7F84 AF	760	XOR	A
7F03	24	120		INC	H	7F85 CDF77E	770	CALL	FUOR
7F04	C9	130		RET		7F88 3E1B	780	LD	A,27
7F05	C5	140	CAR	PUSH	BC	7F8A CDF77E	790	CALL	FUOR
7F06	0608	150		LD	B,8	7F8D 3E41	800	LD	A,65
7F08	C5	160	CAR1	PUSH	BC	7F8F CDF77E	810	CALL	FUOR
7F09	210000	170		LD	HL,0	7F92 3E08	820	LD	A,8
7F0C	19	180		ADD	HL,DE	7F94 CDF77E	830	CALL	FUOR
7F0D	0607	190		LD	B,7	7F97 CD5E7F	840	CALL	TERZ
7F0F	AF	200		XOR	A	7F9A D9	850	EXX	
7F10	CDFE7E	210	INT7	CALL	SETT	7F9B E1	860	POP	HL
7F13	07	220		ADD	A,A	7F9C D9	870	EXX	
7F14	10FA	230		DJNZ	INT7	7F9D F1	880	POP	AF
7F16	CDFE7E	240		CALL	SETT	7F9E C9	890	RET	
7F19	CD777E	250		CALL	FUOR	7F9F	900	END	
7F1C	3AFF7E	260		LD	A,(SETT+1)				
7F1F	0608	270		LD	B,8				
7F21	90	280		SUB	B				
7F22	32FF7E	290		LD	(SETT+1),A				
7F25	C1	300		POP	BC				
7F26	10E0	310		DJNZ	CAR1				
7F28	3E7E	320		LD	A,126				
7F2A	32FF7E	330		LD	(SETT+1),A				





reclame

Vendo Scambio Hard

Vendo penna ottica per Spectrum, L. 30.000 trattabili. Pierluigi Passavanti, corso Langhe 51, 12051 Alba (CN), 0173/33194.

Vendo Spectrum 16K, alimentatore, cavi, manuale, imballo, 50 programmi LJM (anche per 48K) e in Basic, 20 riviste, L. 250.000. Paolo Bondi, via Pellizzi 1, 56100 Pisa, 050/45783.

Vendo Spectrum 48K, presa monitor + monitor Iloz della Commodore, interfaccia joystick due tasti di fuoco programmabile, joystick, penna ottica, registratore, 500 programmi, L. 1.000.000 trattabili. Massimo Mion, via T. Garzoni 3, 31100 Treviso, 0422/65669.

Vendo Spectrum 48K, stampante Alphacom con 8 rotoli carta, manuali inglese e italiano, 300 giochi. Luigi Petrucci, via Sardegna 10, 75100 Taranto, 099/331687.

Solo Campania. Vendo Spectrum 48K, manuale italiano, libri e cassetta gioco scacchi, L. 280.000; oppure scambio tutto con macchina da scrivere elettrica. Francesco Annaruma, via Amedei 23, 84012 Anagni (SA), 081/947323.

Vendo stampante Selkoha GP 50 S con cavo allungato e carta, 1 mese, L. 230.000. Fabrizio Estini, via Palade 65, 39012 Merano (BZ), 0473/34023.

Vendo Spectrum 48 Plus, registratore Philips 66203P con alimentatore, cavi, oltre 100 programmi, manuali italiano e inglese, libri e riviste, tutto nuovo, L. 500.000. Achille Cattaneo, via Milite Ignoto 57, 21027 Ispra (VA), 0332 780139.

Vendo computer Chessmate giocatore di scacchi, istruzioni, pezzi di legno magnetizzati, scacchiera adattatore, 10 livelli di gioco, L. 200.000. Mauro Darida, via Ricciardi 53, 81100 Caserta, 0823 327762.

Vendo Alphacom 32 perfetto stato, imballo, rotoli carta, L. 170.000. Beppe Fasolis, corso Alba 13, 14100 Asti, 0141/53817.

Vendo Currah Speech al miglior offerente. Antonio Farina, via Cancello 33, 81029 Tuoro (CE).

Vendo Spectrum 48K in garanzia, completo di tutto, + listati, giochi, riviste, manuale, L. 300.000. Luigi Pistone, via L. da Vinci 53, 85100 Potenza, 0971/55366.

Vendo QL un mese, manuale, 8 cartucce, gioco scacchi e 4 programmi professionali, L. 900.000. Rodolfo Tusciano, via Sallizi 17, 20025 Legnano (MI), 0331/597054.

Vendo Spectrum 8 mesi ottime condizioni, stampante Selkoha GP 50S, 100 programmi, L. 620.000. Stefano Colombo, via Carotolini 21, 20030 Giussano (MI), 0362/860559 pasti.

Vendo ZX Printer, istruzioni, imballo, rotolo carta, ottimo stato, L. 85.000 (spese comprese). Marco Munari, via Turati 48, 48040 Rivaletta sul Mincio (MN), 0376/853248 pasti.

Vendo stampante Alphacom 32, L. 180.000. Marco Poletto, via Prantenstiel 3, 39100 Bolzano, 0471/971885.

Solo Roma. Vendo Spectrum 48K, L. 300.000; interfaccia per 2 joystick L. 50.000; 400 programmi L. 150.000. Giuseppe Fasulo, via Berni 5, 00186 Roma, 06/7577351.

Vendo Spectrum 48K un anno, tastiera professionale, interfaccia joystick programmabile (senza spinotti), joystick Sunoom, stampante Selkoha GP 50S come nuo-

va, registratore amplificatore Tenkolek, penna ottica con software, libri e 500 programmi, il tutto in apposito mobile con ventola, L. 1.000.000 trattabile. Riccardo Ceruti, via G.B. Piovra 23, 16149 Genova, 010/461661 pomeriggio.

Vendo Atari XL 800 con tavolietta grafica, vinto e mai usato, manuale in italiano, L. 500.000. Marco Cittadini, via Padova 12, 25100 Brescia, 030/347431.

Vendo TI 99, manuale e un libro, 1 cartidge, 100 programmi, praticamente nuovo, L. 200.000. Nello Mecatti, via Stradone 14, 4200 Lantignone (RE), 0522/660110.

Vendo ZX81 16K, tastiera professionale, ZX Printer, cavi, 13 cassette giochi, 5 libri e 26 riviste, 500 programmi, L. 240.000. Alberto Severini, via Olandri 101, 62019 Senigallia (AN), 071/84373.

Per Spectrum vendo Alphacom 32, ZX interfaccia 1, microdrive e 9 cartucce. Marco Pernigoni, via Marocco 17, 20127 Milano, 02/2821860.

Vendo ZX 81, espansione, 6 cassette giochi, 60 programmi vari, cavi, libri, istruzioni, Michele Borotoli, via Monte Cengio 9, 35020 Roncaglia-Ponte S. Nicolò (PD), tel. 718062 pasti.

Vendo espansione 16K per ZX 81, nuova, imballo originale, spedizione contrassegno, L. 30.000. Nicola Pace, via Iro 125, 84100 Salerno, 085/355817 dopo le 22.

Vendo Spectrum 48K, manuale, penna ottica, 100 programmi, L. 290.000. Mario Di Loreto, via A. Saracino 14, 00121 Ostia Lido (RM), 06/5692106.

Vendo Scambio

Soft

Vendono / scambio software per Spectrum (dove non diversamente specificato)

Pierluigi Passavanti, corso Langhe 51, 12051 Alba (CN) 0173/33194.

Beppe Fasolis, corso Alba 13, 14100 Asti, 0141/53817.

Regalo programmi per Spectrum, chiedo solo rimborso spese, francobolli per elenco. Alberto Cruciani, piazza V. Emanuele 136, 00185 Roma.

Alessandro Camela, via Grado 12, 63039 S. Benedetto del Tronto (AP), 0735/86170.

Vendo programma che rende compatibile Vu-file con i microdrives. Giovanni Tagliavini, via Martiri di Gestalonia 14, 43100 Parma, 0521/56767.

Per QL Andrea Galli, via Palagetta 212, 50017 S. Piero a Ponti (FI), 055/899686.

Stefano Fratelli, via Marche 28, 60019 Senigallia (AN), 071/6621155.

Mario Di Loreto, via A. Saracino 14, 00121 Ostia Lido (RM), 06/5692106.

Davide Cantoni, via Mentana 19, 43100 Parma, 0521/73968.

Programmi, listati e mappe. S & F.S., via Di Vittorio 76, 20097 S. Donato Milanese (MI), 02/5272518.

Per QL Gianfranco Balilelli, casella postale 52, 30100 Venezia, 041/28740.

Alessio Forzano, via Nazionale





**TITOLI
IN LINGUA
ITALIANA**

C. A. Street
LA GESTIONE DELLE INFORMAZIONI CON LO ZX SPECTRUM
pag. 136 L. 16.000
ISBN 887700002-3

T. Woods
L'ASSEMBLER PER LO ZX SPECTRUM
pag. 200 L. 18.000
ISBN 887700003-1

G. Bishop
PROGETTI HARDWARE CON LO ZX SPECTRUM
pag. 176 L. 17.000
ISBN 887700005-6

N. Williams
PROGETTAZIONE DI GIOCHI D'AVVENTURA CON LO ZX SPECTRUM
pag. 216 L. 20.000
ISBN 887700007-4

S. Nicholls
TECNICHE AVANZATE IN ASSEMBLER CON LO ZX SPECTRUM
pag. 232 L. 18.000
ISBN 887700010-4

A. Pennell
GUIDA ALLO ZX MICRODRIVE E ALL'INTERFACE 1
pag. 144 L. 16.000
ISBN 887700015-9

G. Kane
IL MANUALE MC68000
pag. 168 L. 16.000
ISBN 887700017-1

NOVITÀ LIBRI
S. Nicholls
GRAFICA AVANZATA CON LO ZX SPECTRUM
pag. 168 L. 18.000
ISBN 887700020-1

J. Jones
IL SUPERBASIS DEL QL
pag. 224 L. 22.000
ISBN 887700028-7

NOVITÀ SOFTWARE
C. A. Street
PROFILE 2 - FOGLIO ELETTRONICO INTEGRATO PER LO ZX SPECTRUM
cassetta L. 24.000
ISBN 887700902-0

S. Nicholls
ROUTINES IN ASSEMBLER PER LA GRAFICA AVANZATA CON LO ZX SPECTRUM
due cassette L. 24.000
ISBN 887700903-9
ACS Software
ZX SPECTRUM MONITOR
cassetta L. 24.000
ISBN 887700905-5

distribuzione in libreria:
Messaggerie Libri S.p.A.
Via Giulio Carcano, 32
20141 MILANO MI
tel. 02 8438141-8467341, telex 310672 MESSIT I

McGraw-HILL BOOK COMPANY GmbH
Ladenanbogen 136
D-2000 Hamburg 63
REPUBBLICA FEDERALE TEDESCA
tel. 041 770 5382081, telex 2164048 MHBC D



216, 96063 Giolosa Maree (ME),
0941/301117.

Solo Toscana. Marco Simonetti,
viale Montegrappa 35, 50047 Prato
(FI), 0574/593778 pasti.

Zona Milano. Darlo Passadore,
viale Vigliani 19, 20148 Milano,
02/4989501.

Zona Roma. Maurizio Delle Frate,
via G. Partini 15, 00169 Roma,
06/265280 pasti.

Zona Piacenza. Edo Mars, via
Gonzaga 22, 29100 Piacenza,
0523/71467.

Maurizio Turco, via Casati 26,
20124 Milano, 02/6706435.

Guido Singarella, via Conte Ros-
so 56, 85100 Campobasso,
0874/94764.

Fabrizio Martano, via don Sturzo
7, 58100 Grosseto, 0564/492806.

Svendo anche in blocco ca 1000
programmi. Doriana Orlandi, via
Qualmodo 6/C, 46023 Gonzaga
(MN), 0376/588555.

Paolo De Rosa, via Montepelle-
grino 144, 90142 Palermo,
091/545384.

Zona Bergamo, da settembre.
Andrea Taricco, via Bologna 10,
24100 Bergamo, 035/218239.

Pietro Cardetta, vico Rosmini 3,
74017 Mottola (TA).

Per Commodore, IBM, Apple. Tiz-
iano Silvestri, via Tuscolana
1252, 00174 Roma, 06/7491440
9-13, 16-20.

Solo Roma. 400 programmi, L.
150.000. Giuseppe Fasulo, via Ber-
ni 5, 00185 Roma, 06/7577351.

Davide Pintus, via Nuoro 3,
09042 Mandas (CA).

Compro Cerro Varie

Compro interfaccia 1 e microdr-
ve, preferibilmente Emilia-
Romagna-Marche. Pietro Di Pilato,
via Raggi 26, 47100 Forlì,
0543/69648 dopo le 20.

Sinclub Gordisa, c/o Stefano Pie-
monti, viale Virgilio 44, 34170 Gor-
zia, 0481/82174.

QL User Club cerca soci in tutta
Italia, c/o Roberto Ghezzi in via
Volontari del Sangue 202, 20099 Se-
sto S. Giovanni (MI), 02/2485511.

Bug Club, che aderisce al Sin-
club Italia, L. 3500 catalogo su na-
stro. Cambiamo il miglior software
con accessori hardware e stam-
panti purché in ottime condizioni.

c/o Balano, via Battisti trav. priva-
ta 11, 80059 Torre del Greco (NA),
081/8816572.

Realizzo qualsiasi programma
di archivio data base per Spec-
trum. Gianpaolo Gentili, via Turati
10, 10024 Moncalieri (TO),
011/6407195.

Cerco possessori di QL per fon-
dare un club. Umberto Mancini, via
Mezzetti 25, 74100 Taranto,
099/338735.

Cerco possessori di adventures.
Cristian Coppa, via Marconi 27,
40139 Bologna, 051/543990.

Vendo ricetrasmittitore CB La-
fayette 1800 20 canali AM-FM-
USB-LSB 4 bande + echo-tristar,
usati pochissimo, L. 300.000, o
scambio con Commodore 64 o
stampante per Spectrum non ter-
mica. Romano Guldi, via Cantoreg-
gio 11, 21100 Masnago (VA),
0332/223482.

Cerco libri e periferiche per
Spectrum in cambio offro pro-
grammi; vendo riviste per compu-
ter a metà prezzo. Gianpaolo Gen-
tili, via Turati 10, 10024 Moncalieri
(TO), 011/6407195.

Cerco utenti QL per scambio
software e eventuale club zona
Genova. Alessandra Catorcini, via
Crocco 31 F. 6, 16122 Genova,
010/810283.

Cerco urgentemente i program-
mi: Bruce Lee, Rocky Oror Show,
Gremlins, The Duke of Hazard,
Broad Street, con istruzioni e a
buon prezzo. Roberto Ravani, via
Cimabue 32, 20052 Monza (MI),
039/834367.

Per QL, possibilmente zona To-
rino. Roberto Beretta, corso Agnel-
li 130, 10137 Torino, 011/359801.

Compro Sinclair QL, in cambio
cedo: Spectrum 48K, registratore
dedicato, 100 programmi, manuali-
stica, rivista, eventuale congua-
glio. Paolo Marinelli, via C. Cam-
pagna 9, 37131 Verona, 045/520170
ore 19-20,30.

Cerco possessori QL per scam-
bio informazioni. Marco Poletto,
via Pfannenstiel 3, 39100 Bolzano,
0471/971887.

Scambio 300 programmi a scelt-
a con stampante Seikosha GP
505 o altra compatibile con Spec-
trum, oppure con microdrive, o con
interfaccia 1. David Pintus, via Nuoro
3, 09042 Mandas (CA).

Cerco Fighter Pilot e Pole Posi-
tion; scambio libri fotocopiati. Pie-
tro Cardetta, vico Rosmini 3, 74017
Mottola (TA).

Zona Piacenza: Vendo manuale
Italiano Pascal HiSoft HPAT. Edo
Mars, via Gonzaga 22, 29100 Pia-
cenza, 0523/71467.

Cerco monitor non a presa
RGB, anche a fosfori; cerco Seiko-
sha GL 505 o buon prezzo. Mario
Di Loreto, via Saracino 14, 00121
Ostia Lido (Roma), 06/5992106.

Vendo videogiochi tascabili a
prezzo da concordare. Pierluigi
Passavanti, corso Langhe 51,
12051 Alba (CN), 0173/33194.



SINCRIF
COMPUTER

Bastimento
16/48 didattico
Disassembla
English
Corso LM
Script video

Bioritmi

Test QS

Il lungo ritorno (5)



Systems

Responsabile: Aquilino Pizzarello
C/Donna Ignazia, 200017162
Ind. di Mod. 1 01771394 s.r.l.
50100 Carr. S. Antonio 100/5

anno 2 - n. 5 - Maggio-Giugno 1985 - Lire 8.500

In omaggio
il secondo libro
porta-nasrri





32 BIT

32 BIT

16 BIT

16 BIT

8 BIT

8 BIT



SINCLAIR QL: AL VERTICE DELLA NUOVA GENERAZIONE

Sinclair QL rivoluziona il mondo dei computer, perché combina le dimensioni di un home con la potenza e le capacità di un mini.

QL è l'unico computer, nella sua fascia, ad impiegare il microprocessore a 32 bit, quando gli altri si fermano a 8 oppure 16.

La sua portentosa memoria è di 128 KRAM espandibile a 640.

I quattro programmi applicativi, già incorporati, sono immediatamente utilizzabili e superano, in qualità, il software dei microcomputer esistenti.

Ha la possibilità di multitask e può essere inserito in reti di comunicazione.

Grazie ai due microdrive e al software incorporati, Sinclair QL, nella sua confezione originale, è già pronto per l'uso: basta collegarlo ad un video.

E pensare che tutta questa tecnologia pesa meno di due chili e trova spazio in una normale 24 ore.

Un computer così non poteva che essere Sinclair.

sinclair

Distribuzione esclusiva: GBC Divisione Rebit.

Tutti i prodotti Sinclair, distribuiti da GBC Divisione Rebit, sono corredati da regolare certificato di garanzia italiana.

